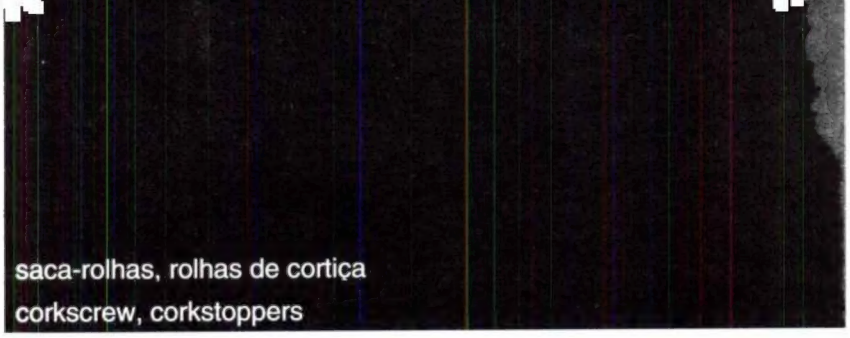
TÉCNICA
revista de Engenharia - 92



saca-rolhas, rolhas de cortiça
corkscrew, corkstoppers

A. Sousa e Brito
Departamento de Engª Mecânica, IST, UTL
M. Amaral Fortes
Departamento de Engª de Materiais, IST, UTL

DETERMINAÇÃO EXPERIMENTAL DAS FORÇAS DE INTRODUÇÃO E REMOÇÃO DUM SACA-ROLHAS

Resumo

Mediu-se a força axial e o binário necessários à introdução de saca-rolhas (SR) em rolhas de cortiça, em função da altura de penetração. Efectuaram-se dois tipo de ensaios. Num deles, utilizou-se uma máquina de ensaios mecânicos com um dispositivo que permite a sincronização entre a rotação e o avanço do SR. Nos outros ensaios utilizaram-se balanças e dinamómetros. Foram testados dois tipos de SR. Os resultados obtidos são bem explicados pela análise efectuada em artigo anterior [1], mostrando uma variação linear com a profundidade, quer da força quer do binário, com os declives previstos. Efectuaram-se também medidas na remoção do SR do interior da rolha.

Abstract

The axial force and the torque required to introduce a corkscrew in a cork stopper were measured as a function of penetration depth. Two types of experiments werw conducted. In one, a mechanical testing machine was used with a device for syncronizing the rotation angle with the penetration depth. Other measurements were done with balances and dynamometers. Two types of corkscrews were used. The experimental results can be explained by the analysis undertaken in [1], and show a linear variation with penetration depth, both of the axial force and of the torque, the slopes being in good agreement with the predictions. Measurements were also carried out for the removal of the corkscrew out of the cork.

Introdução

A geometria e a mecânica do saca-rolhas (SR) foram descritos em artigo anterior [1]. A introdução dum SR numa rolha de cortiça requer a aplicação de um binário, T, e de uma força axial, F, que aumentam linearmente com a profundidade de penetração, H:

$$T = T_0 + k_T H \tag{1}$$

$$F = F_0 + k_F H \tag{2}$$

T_0 e F_0 correspondem à penetração da extremidade aguçada do SR. A altura ou profundidade de penetração, H, é medida a partir do fim da penetração da extremidade. A análise elaborada em [1] prevê, para o caso dos SR tradicionais, formados por um arame enrolado em hélice,

$$k_T = \frac{dT}{dH} = \mu\bar{\sigma}\frac{2\pi r_0^2}{p}P_0 \tag{3}$$

$$k_F = \frac{dF}{dH} = \mu\bar{\sigma}P_0 \tag{4}$$

em que p é o passo do SR, r_0 o raio médio das hélices que formam o SR e P_0 é o perímetro da secção recta, isto é, da secção perpendicular à hélice de raio r_0 (nos SR tradicionais $P_0=\pi D$, em que D é o diâmetro do arame); $\bar{\sigma}$ é o valor médio da pressão sobre a superfície do SR (tensão de compressão na cortiça) e μ o coeficiente de atrito de escorregamento entre o SR e a cortiça. O valor de $\bar{\sigma}$ é um pouco superior a 1 MPa [1].

As equações (1) e (2) também são aplicáveis a SR com outras geometrias, mas as grandezas k_T e k_F têm uma relação mais complicada com a geometria do SR. Combinando (3) e (4) obtém-se para a relação entre os declives de $T(H)$ e $F(H)$:

$$\frac{k_T}{k_F} = \frac{dT/dH}{dF/dH} = \frac{2\pi r_0^2}{p} \tag{5}$$

Num outro tipo de SR, com um núcleo central levemente cónico, a variação de T e de F com H é presumívelmente quadrática [1], mas o coeficiente do termo H^2 é pequeno visto que a conicidade do núcleo é sempre muito pequena.

Nos SR comerciais, a extremidade aguçada é muito curta (≈3 mm) e por isso os termos F_0 e T_0 nas equações (1) e (2) podem na prática desprezar-se quando se

Original submetido para publicação em 10 de Maio de 1992

Após extracção da rolha, o SR é retirado da mesma, o que implica a aplicação de um binário e de uma força de sentidos contrários aos da introdução. Em [1] prevê-se que os valores absolutos respectivos sejam inferiores aos da introdução, para o mesmo *H*, embora variando ainda linearmente com *H*.

Neste artigo descrevem-se os resultados de experiências destinadas a medir, ao que sabemos pela primeira vez, o binário e a força de introdução e remoção do SR. Com os resultados obtidos testaram-se as previsões efectuadas em [1], nomeadamente as eqs. 1-5.

Parte experimental

Utilizaram-se dois SR de aço, cujas características geométricas estão indicadas na tabela 1. Um dos SR é do tipo tradicional (arame enrolado) e o outro é um SR com núcleo cónico.

tabela 1
Características geométricas do SR tradicional de arame e do SR de núcleo cónico

	Saca-rolhas	
Dimensões (mm)	Arame	Núcleo cónico
Passo (p)	9,2	9,0
Raio médio (r_n)	3,3	--
Comprimento do SR	40	40
Raio exterior	4,7	4,5
Diâmetro do arame (D)	3,0	--
Raio do núcleo na zona média	--	3,0
Ângulo de conicidade do núcleo	--	3,7°

Foram efectuados dois tipos de ensaios de medida de *F* e *T* em função de *H*. Nos ensaios do 1º tipo utilizou-se uma máquina Instron de ensaios mecânicos com acessório para ensaios de torção. Os ensaios do 2º tipo foram ensaios expeditos recorrendo-se a balanças e dinamómetros.

Nas medidas efectuadas com a máquina Instron, os provetes ensaiados foram rolhas de cortiça de geometria normalizada (24 mm de diâmetro e 45 mm de altura), montadas em cápsulas tubulares de ebonite com diâmetro interior igual ao diâmetro normalizado dos gargalos de garrafas de vinho (18 mm). Nestas condições, o aperto da cortiça é presumivelmente idêntico ao que ocorre nas garrafas. As cápsulas contendo as rolhas foram por sua vez montadas num dispositivo metálico fixado ao travessão móvel da máquina de ensaios, através da célula de carga. A figura 1 contém um esquema do dispositivo experimental.

A dificuldade da medida do binário e da força reside no facto de ter de haver uma sincronização entre o movimento de rotação e o movimento de translação paralela ao eixo do SR: uma rotação completa está associada a uma translação igual ao passo do SR. Para obter esta sincronização, a máquina de ensaios (do tipo electromecânico) foi provida de um sistema apropriado, ligado à roda principal de engrenagem, ao qual foi fixado o SR (figura 1). A célula de carga de torção utilizada na medida dos binários de introdução e de remoção do SR tinha uma capacidade elevada (200 Nm) o que não permitiu determinações de boa sensibilidade (os momentos máximos medidos foram de 1,6 Nm). A velocidade do travessão foi de cerca de 10 mm/min, sincronizada, como se disse, com a velocidade de rotação do SR, ou seja, cerca de 1 rpm, uma vez que $p \approx 10$ mm. Com o mesmo dispositivo, mas utilizando uma célula de carga tracção-compressão (capacidade 10^4 N), mediu-se, em outros ensaios, a força axial de introdução e de remoção do SR.

Nestes ensaios com a máquina Instron foram obtidos registos das curvas *T(H)* e *F(H)* de introdução e de remoção do SR, de que se mostra um exemplo na figura 2. Em

todos os casos, as curvas apresentam um forte serrilhado, que tem a ver com a heterogeneidade da cortiça (serrilhado semelhante observa-se nas curvas *F(H)* de penetração de pregos na cortiça e em outros materiais similares [2]).

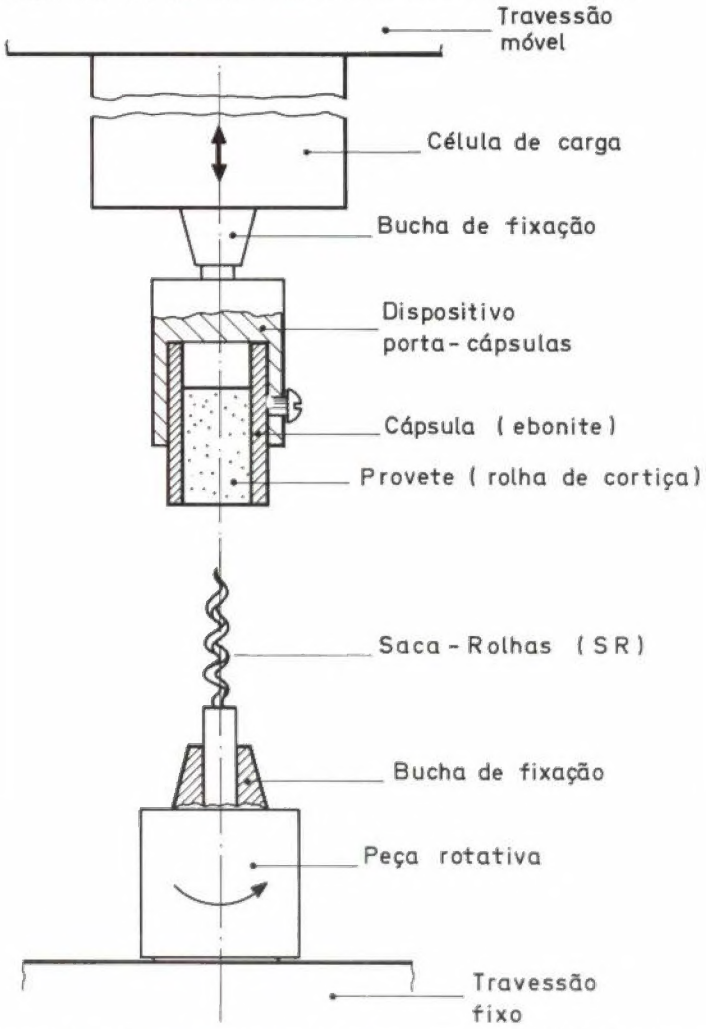


figura 1 Esquema do dispositivo montado na máquina de tracção para ensaios de medida do binário e da força axial, na introdução/remoção de SR.

Numa segunda série de experiências foram efectuadas medidas do binário e da força axial de introdução e remoção dos mesmos SR em rolhas montadas nas cápsulas atrás descritas bem como em rolhas colocadas em garrafas de vidro comerciais. A força axial foi medida colocando a cápsula ou a garrafa rolhada sobre uma balança dinamométrica (sensibilidade 5 N) com a qual se mediu a força necessária para introduzir manualmente o SR; os valores foram medidos de meia em meia rotação completa, a partir da penetração da 2ª espira, por razões que têm a ver com a necessidade de fixação do SR e com o equilíbrio do conjunto. O binário de introdução do SR foi medida por um dinamómetro de mola (capacidade 25 N, sensibilidade 1 N) aplicado a uma haste transversal fixada ao SR (braço da força: 13 cm). Nestas experiências, a cápsula ou a garrafa estavam colocadas sobre a balança utilizada nos ensaios de medida da força axial, o que tornou possível o controlo da força axial que tem de ser aplicada simultaneamente, e cujos valores tinham sido previamente medidos. Também neste caso as medidas foram efectuadas em cada meia rotação a partir da penetração da 2ª espira.

Por último, mediram-se também, em ensaios expeditos, os valores do binário e da força axial na introdução e na remoção de SR em rolhas não submetidas a nenhum aperto, isto é, fora dos gargalos.

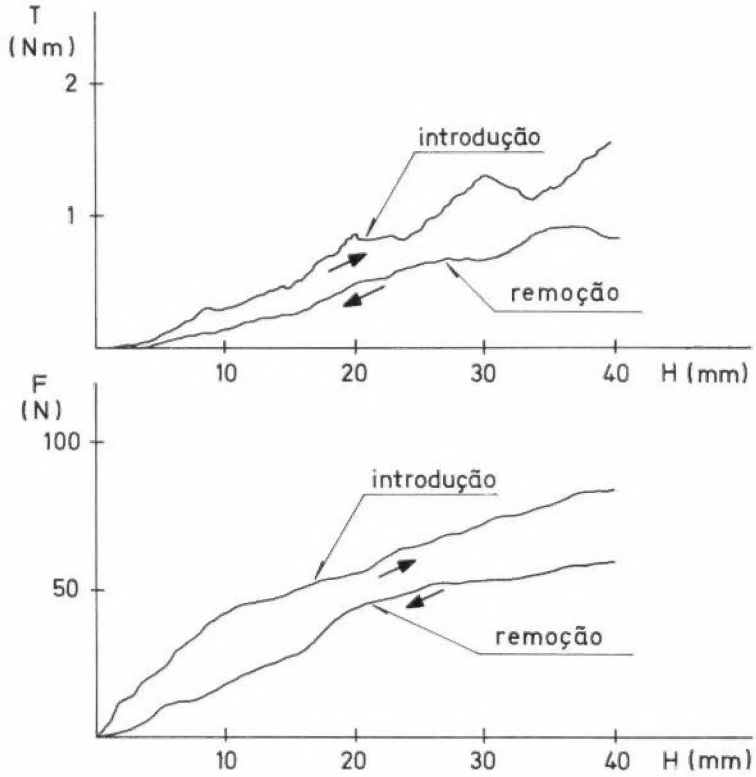


figura 2 Exemplo de curvas T(H) e F(H) obtidas em ensaios de introdução e remoção de SR em rolhas de cortiça introduzidas em gargalos.

Resultados e discussão

As curvas *T(H)* e *F(H)* na introdução e remoção do SR tradicional (figura 2) são do tipo previsto pelas equações (1) e (2), isto é, lineares, não obstante o forte serrilhado que apresentam. Como se previu em [1], o binário e a força axial são menores na remoção (com a rolha no gargalo) do que na introdução (cerca de 20% menores) e os declives são também menores.

Os valores experimentais médios dos declives de *T(H)* e *F(H)* obtidos em 2 ensaios com a máquina Instron e em 6 ensaios expeditos, na introdução e remoção do SR tradicional, são apresentados na tabela 2.

tabela 2
Valores experimentais médios do binário e força axial na introdução e remoção do SR tradicional de arame para rolhas no gargalo.

Ensaio	*dT/dH* (N)	*dF/dH* (Nm^{-1})	*dT/dF* (mm)
Máquina Instron			
-Introdução	37,1	2250	16,5
-Remoção	31,0	1880	16,5
Ensaios expeditos			
-Introdução	44,9	2647	16,9
-Remoção	32,1	1814	17,7

Estes valores referem-se a rolhas dentro de gargalos e cápsulas, tendo-se observado valores mais elevados no caso das rolhas metidas em cápsulas. É de notar que a dispersão de valores é relativamente elevada, o que se deve imputar, não aos métodos experimentais, mas sim à variabilidade do comportamento das próprias rolhas. Os desvios padrão nos 6 ensaios expeditos foram os seguintes: 9 N para *dT/dH* e 460 Nm^{-1} para *dF/dH*.

O quociente dos declives de *T(H)* e *F(H)* na introdução do SR tradicional tem portanto o valor experimental médio:

$$\frac{dT}{dH} \Big/ \frac{dF}{dH} = 17 \times 10^{-3} \text{ m} \qquad (6)$$

Utilizando os valores da tabela 1 obtém-se

$$\frac{2\pi r_0^2}{p} = 7{,}4 \times 10^{-3} \text{ m} \qquad (7)$$

Há portanto uma diferença de cerca de um factor 2 relativamente à previsão da equação 5. Recorde-se que esta equação é aproximada, pressupondo uma pressão uniforme sobre o SR [1]. O facto do valor experimental ser superior poderá indicar que a pressão é maior do lado exterior do que do lado interior das espiras do SR. Também se obtém

$$\frac{2\pi r_0^2}{p} P_0 = 7{,}0 \times 10^{-5} \text{ m}^2 \qquad (8)$$

o que permite obter, das equações (3) e (4), valores de $\mu\bar{\sigma}$ entre 0,3 e 0,6 MPa. Como $\bar{\sigma}$ deve ser próximo, mas superior a 1 MPa, conclui-se que o coeficiente de atrito μ tem um valor próximo de 0,5.

As curvas *T(H)* e *F(H)* obtidas com o SR de núcleo cónico são semelhantes às relativas ao SR tradicional, não se tendo detectado desvio em relação à linearidade. Os declives medidos, correspondentes aos ensaios com a máquina Instron e aos ensaios expeditos, são apresentados na tabela 3, não diferindo muito dos obtidos com o SR tradicional (tabela 2). Para uma profundidade de penetração de 40 mm, correspondente às 4 espiras que habitualmente se introduzem na rolha, os valores máximos (finais) do binário e força axial são, para ambos os SR, à volta de 1,6 Nm e 90 N, respectivamente.

tabela 3
Valores experimentais médios do binário e da força axial na introdução e remoção do SR de núcleo cónico para rolhas no gargalo.

Ensaio	*dT/dH* (N)	*dF/dH* (Nm^{-1})	*dT/dF* (mm)
Máquina Instron			
-Introdução	30,7	-	-
-Remoção	29,4	-	-
Ensaios expeditos			
-Introdução	45,7	2444	18,7
-Remoção	27,0	1820	14,8

Finalmente na tabela 4 indicam-se os valores medidos de *dT/dH* e *dF/dH* com os dois tipos de SR, mas utilizando rolhas não apertadas, isto é, não introduzidas em gargalos. Verifica-se que o aperto da rolha provoca valores mais elevados das forças (factor 1,1 a 1,2), o que pode ser explicado pelo maior valor da pressão $\bar{\sigma}$ que a cortiça exerce sobre o SR.

tabela 4
Valores experimentais médios do binário e da força axial na introdução e remoção do SR para rolhas fora do gargalo.

Ensaio	dT/dH (N)	dF/dH (Nm^{-1})	dT/dF (mm)
SR tradicional			
-Introdução	33,2	1578	21,0
-Remoção	26,4	1295	20,4
SR de núcleo cónico			
-Introdução	36,2	2098	17,3
-Remoção	24,4	1438	17,0

Agradecimentos

Os autores agradecem ao Laboratório Nacional de Engenharia Civil e ao Laboratório de Cortiça e Produtos Resinosos da Estação Florestal Nacional, o apoio na realização dos ensaios. Em especial agradece-se a colaboração prestada pelos Srs. Eng. Martins de Brito e José António Varanda.

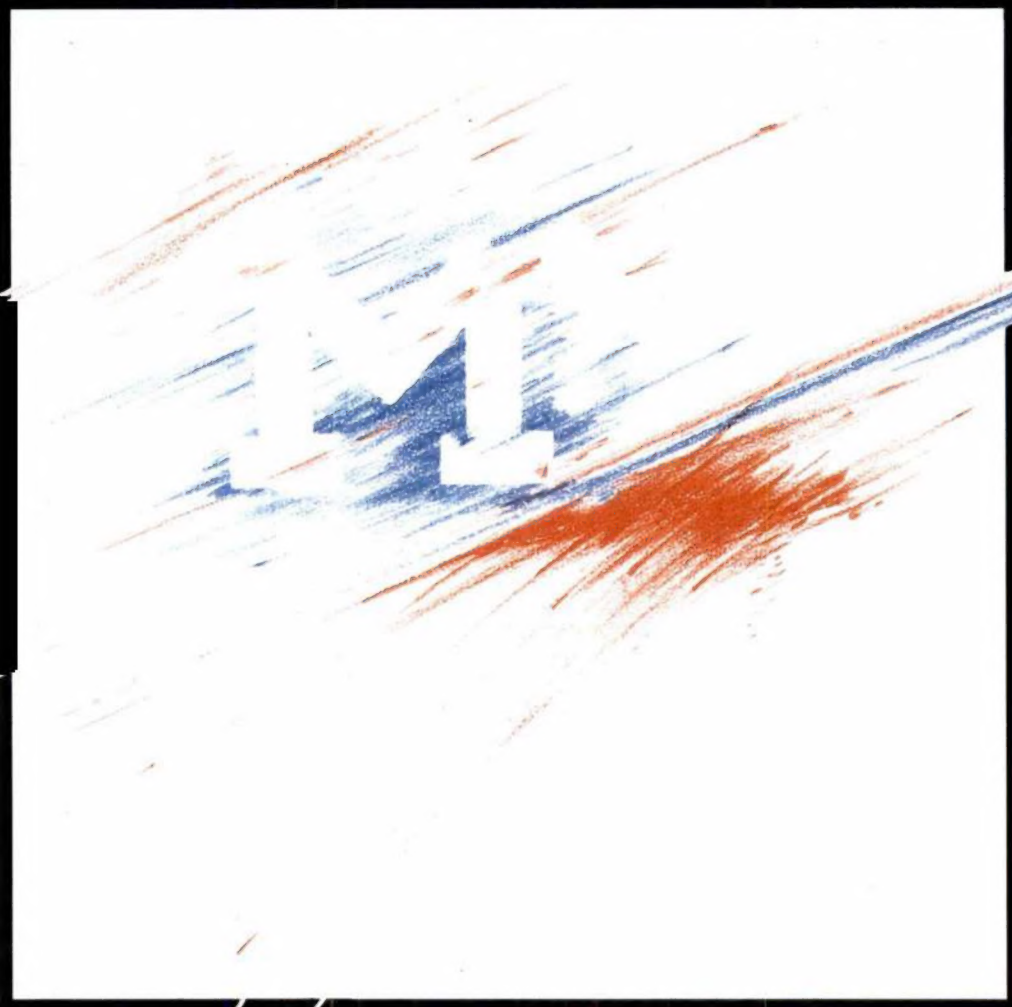
Referências

[1] Fortes, M. Amaral (1993). A geometria e a mecânica do saca-rolhas, *Técnica*, número único de 1992, artigo precedente neste número.

[2] Vaz, M. Fátima e Fortes, M. Amaral (1991). The identation and nailing of cellular materials in Mechanical Properties of Porous and Cellular Materials, *Mat. Res. Soc. (USA) Symposium Proceed.*, 207, 41-59.

Sobre os Autores

A. Sousa e Brito é Professor Auxiliar do Departamento de Engenharia Mecânica e M. Amaral Fortes é Professor Catedrático do Departamento de Engenharia de Materiais, ambos do Instituto Superior Técnico, Universidade Técnica de Lisboa



Estamos a crescer com Lisboa

M • PROJECTO • EM • MOVIMENTO •

METROPOLITANO DE LISBOA

COLORFOTO
fotografia e cinema

de: BARREIROS DA SILVA, LDA.
MATERIAL PARA FOTOGRAFIA E CINEMA

Nikon

Kodak

GITZO

Canon

FUJI

MATERIAL FOTOGRÁFICO
AMADOR • PROFISSIONAL
REVELAÇÃO DE DIAPOSITIVOS
FOTOGRAFIA A CORES E AMPLICÓPIAS
A CORES ATÉ FORMATO 20 × 30
EM 1 HORA

RUA VISCONDE DE SANTARÉM, 75-C 352 51 66

PRAÇA DE ALVALADE, 2D - 1700 LISBOA - TELEF. 793 24 75

HP

HIDROTÉCNICA PORTUGUESA

Barragem El Whada – Marrocos

Planeamento de recursos
Hidráulica agrícola e urbana
Aproveitamentos hidroeléctricos
Estruturas especiais
Vias de comunicação
Engenharia litoral e portuária
Redes e instalações de gás
Ordenamento do território e edificações
Gestão e fiscalização de empreendimentos

HIDROTÉCNICA PORTUGUESA
CONSULTORES PARA ESTUDOS E PROJECTOS, LDA. – Edifício HP – Rua da Guiné – 2685 SACAVÉM – Telex: 18180 HP Lx – Fax: 941 86 65

redes neuronais, perceptrões multi-camada, retropropagação, redes recorrentes

neural networks, multilayer perceptrons, backpropagation, recurrent networks

Luís B. Almeida
Dep. Engª Electrotécnica e de Computadores, IST, UTL

AN INTRODUCTION TO MULTILAYER PERCEPTRONS

Resumo

A *retropropagação* é provavelmente a mais rápida e usada regra de aprendizagem para redes neuronais. Este artigo apresenta o algoritmo de retropropagação de uma forma tutorial. Começa-se por uma breve introdução ao perceptrão de Rosenblatt, e ao *adaline* de Widrow e Hoff. Em seguida, são estudados os perceptrões multicamada não recorrentes (*feedforward*), com aprendizagem pela retropropagação. A extensão da retropropagação a redes recorrentes não sequenciais é apresentada em seguida; é dada uma derivação relativamente simples da retropropagação recorrente. Finalmente, são analisadas as redes recorrentes sequenciais.

Abstract

Backpropagation is probably the fastest and most widely used learning rule for neural networks. This paper presents the backpropagation algorithm in a tutorial form. Is starts by a short introduction to Rosenblatt's perceptron, and to Widrow and Hoff's adaline. Then, multilayer feedforward perceptrons with backpropagation learning are studied. The extension of backpropagation to recurrent, non-sequential networks is presented next; a relatively simple derivation of the recurrent backpropagation rule is given. Finally, recurrent sequential networks are analysed.

Introduction

Artificial neural networks have been an object of much interest in recent years. Among the reasons for this interest are their learning capabilities, their potential for massively parallel processing and their fail-soft properties. Two of the earliest neural network structures with learning capabilities were the *perceptron*, introduced in the late fifties by Rosenblatt [1] and the *adaline*, introduced shortly thereafter by Widrow and Hoff [2] (calling them networks is slightly abusive, since in essence each of them consists of a single neuron-like unit). Being able to learn from experience, the perceptron created a great deal of interest, and many expectations were put on it as the basis for more complex learning networks (the adaline is a linear system, as we shall see, and its limitations were easy to recognise). In 1969 Minsky and Papert published a book which become a classic [3], in which they analysed in great depth the capabilities of the single-unit perceptron, finding that it also had very severe limitations. These limitations could be removed by considering multi-unit, multilayer perceptrons, but finding a way to perform learning in these multilayer structures proved far more difficult than initially expected. Only in recent years were algorithms found for performing learning in multilayer networks. Among these, the one that is most widely used, and that can probably be considered as the most direct extension of the learning algorithms of the perceptron and the adaline, is the so-called *backpropagation learning rule*. In the following sections we will first briefly present Rosenblatt's perceptron and the adaline, and we will then introduce the backpropagation learning rule for feedforward multilayer systems (i.e. multilayer systems with no feedback). We will proceed by extending backpropagation to continuous-time recurrent systems (i.e. systems with feedback connections) in non-sequential applications. Finally we will discuss the application of backpropagation to discrete-time recurrent systems, in sequential applications.

Rosenblatt's perceptron

The essential part of the perceptron, as introduced by Rosenblatt, is a single neuron-like unit, as depicted in figure 1. It computes a weighted sum of its inputs x_i using weights a_i, plus a bias term a_0, according to

$$s = \sum_{i=1}^{I} a_i x_i + a_0$$

The bias term can be incorporated into the summation symbol, by defining a dummy input x_0, which has always a value of 1. Then,

$$s = \sum_{i=0}^{I} a_i x_i$$

Original submetido para publicação em 10 de Fevereiro de 1992

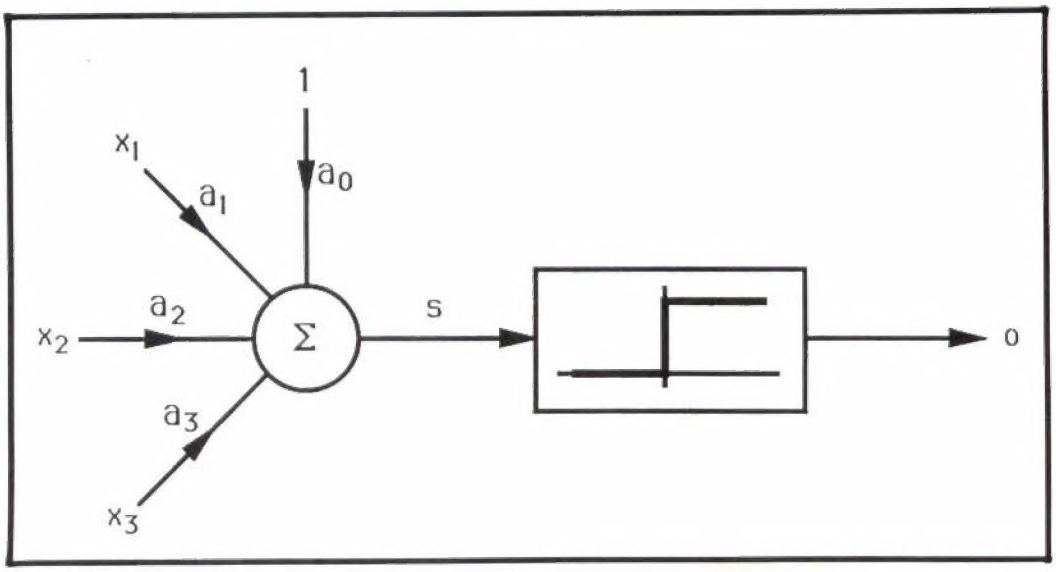


figure 1 Rosenblatt's perceptron

This sum is passed through a step function, whose output is

$$o = \begin{cases} 1 \text{ if } s > 0 \\ 0 \text{ if } s \leq 0 \end{cases}$$

The inputs of the perceptron can be any desired signals. For example, they can come from the pixels of an image, but they can just as well be, say, the heartbeat rate, breathing rate, blood pressure values, etc. in a patient monitoring system. They can also be binary logic values, or some input variables can be binary, and other ones analog.

Since the perceptron has a binary output, we can think of it as performing a classification of the input patterns into two classes, one corresponding to those patterns that produce an output of 0, and the other to those that produce an output of 1. The function, or classification, performed by the perceptron can be changed by modifying the weights a_i. A *learning rule* is an algorithm for modifying the weights in such a way that the perceptron will implement some desired function.

Rosenblatt's learning algorithm is a form of supervised learning: the way to show the perceptron what is the function that it is desired to perform, is by giving it examples of input patterns and of the corresponding desired outputs; the "teacher" is the person (or system) that tells it what is the desired output for each pattern that is presented at the input.

The learning algorithm itself is quite simple: each iteration consists of presenting an input pattern x_i (i=1,..., l), observing the output o produced by the perceptron, and then performing one of the following steps, depending on the values of the output o and of the desired output d:

---

1 Rosenblatt's theorem assumes two additional hypotheses. The set of patterns being considered must be finite, and the sequence of patterns presented to the perceptron cannot be completely arbitrary: every pattern must recur if we continue the training for a sufficiently large number of steps.

1) if o=0 and d=1, update the weights according to $a_i^{(n+1)} = a_i^{(n)} + \eta x_i \quad (i = 0,\ldots,I)$

   where $a_i^{(n+1)}$ are the updated values, and n is the iteration number

2) if o=1 and d=0, update according to $a_i^{(n+1)} = a_i^{(n)} - \eta x_i \quad (i = 0,\ldots,I)$

3) if o=d, leave the weights unchanged $a_i^{(n+1)} = a_i^{(n)} \quad (i = 0,\ldots,I)$

In this algorithm, η is any positive constant, and there are no constraints on the initial values of the weights. Rosenblatt showed that, if the desired classification is feasible with some set of weights, this algorithm will find an appropriate set of weights in a finite number of steps[1]. This means that the perceptron will "learn" to make the desired classification after a finite number of presentations of examples of that classification.

Note that this learning rule can be expressed in a more compact form, by saying that the weights must be updated according to

$$a_i^{(n+1)} = a_i^{(n)} - \eta x_i (o - d)$$

or

$$a_i^{(n+1)} = a_i^{(n)} - \eta x_i e \qquad (1)$$

where e=o-d is the output error, which can only take the values -1, 0 and 1. These update equations can be used for all four combinations of values of o and d.

An intuitive explanation of why Rosenblatt's learning rule works comes from the observation that in case 1 above, for example, the input sum after the weight update is

$$\sum_{i=0}^{I} a_i^{(n+1)} x_i = \sum_{i=0}^{I} a_i^{(n)} x_i + \eta \sum_{i=0}^{I} x_i^2 > \sum_{i=0}^{I} a_i^{(n)} x_i$$

that is to say, if for a certain pattern, the output is 0 and the desired output is 1, the weights are changed in such a way that the input sum is increased. After a certain number of presentations of that pattern, this sum will eventually become positive, and the output of the perceptron will take the correct value. A similar reasoning can be made for case 2, where the sum is decreased, eventually making the output change from 1 to 0. Of course, this reasoning did not take into account the fact that interspersed presentation of other patterns will also change the weights, and may in some cases tend to counter the changes produced by the pattern that we are considering. However, it can be shown that the weight modification prescribed by Rosenblatt's rule is, in a certain sense, the smallest one that would achieve the same amount of variation in the input sum, and thus, intuitively, it tends to disturb the behaviour for other patterns as little as possible.

As previously mentioned, the perceptron created a great deal of enthusiasm when it was introduced, because of its ability to learn from experience. However, an important condition in Rosenblatt's proof is that the desired classification be feasible with some set of weight values (for unfeasible classifications, application of the rule will result in endless oscillation of the weights). The feasibility limitation turned out to be much more severe than was initially realised. In 1969, Minsky and Papert published a book [3] in which they analysed in great depth the capabilities and limitations of Rosenblatt's perceptron. Their work is by far too long and too complex to be summarised here, but the basic source of the limitation is easy to understand. Consider each input pattern as a point in an I-dimensional space, whose coordinates are the values of the inputs x_i

(i=1,..., l). The boundary between the points that produce an output of 0 and those that produce an output of 1, corresponds to those points for which the input sum is exactly equal to zero:

$$\sum_{i=0}^{l} a_i x_i + a_0 = 0$$

This is the general equation of a hyperplane in l-dimensional space, and thus the feasible classifications are those that can be expressed by hyperplane boundaries. Simple examples of feasible classifications are the AND and OR logic functions, while the XOR is an example of an unfeasible classification. These functions are depicted in figure 2, where possible classification boundaries are shown. Note that, since the input space is bidimensional, hyperplanes correspond to straight lines, in this case. There is no single straight line that can separate zeros from ones in the XOR case.

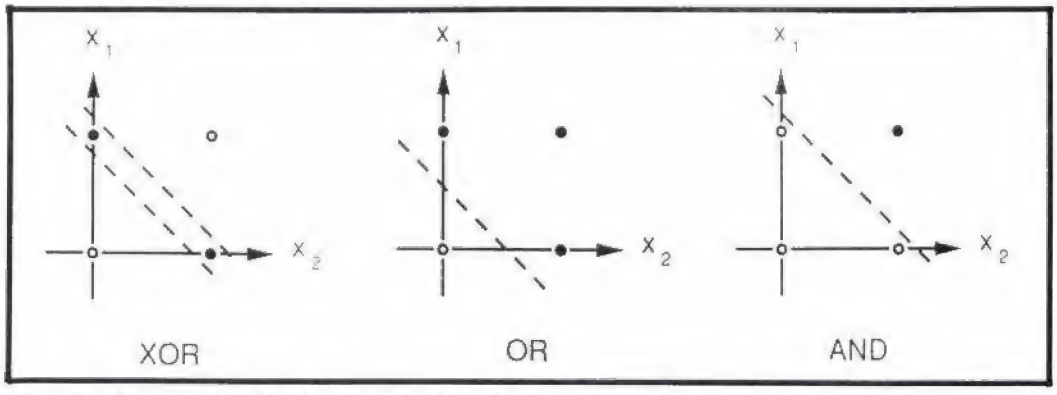


figure 2 Examples of logical functions that are feasible, and unfeasible, by Rosenblatt's single layer perceptron. A 1 at the output is represented by a black dot, a a 0 by a white dot. AND and OR are feasible, XOR is unfeasible.

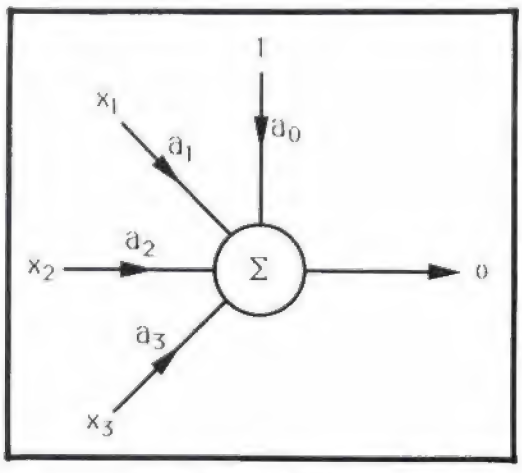


figure 3 The adaline.

As the reader can possibly imagine, most useful problems involve non-hyperplane boundaries, and thus cannot be solved by Rosenblatt's perceptron. However, the perceptron has still frequently been used as a classifier (often under the name of *linear classifier*), its weights being computed by variants of Rosenblatt's procedure or by alternate methods.

The delta rule

A learning rule which is almost contemporary to Rosenblatt's, is the so-called *delta rule*, introduced by Widrow and Hoff [2]. They considered a linear device that corresponds only to the first stage of the perceptron (see figure 3). This was called *adaline* (for ADAptive LINear Element). While the perceptron produces a binary output, the adaline produces an analog one:

$$o = \sum_{i=0}^{I} a_i x_i$$

Like we did for the perceptron, assume that we have a set of training patterns, for which we know the desired outputs. For convenience, represent the patterns by vectors $\boldsymbol{x}^k$ (we will use bold letters to denote vectors). Denote by d^k the corresponding desired outputs. Also, denote by o^k the output produced by the adaline, with a given set of weights, when $\boldsymbol{x}^k$ is presented at the input. A measure of the performance of the adaline is the total quadratic error

$$E = \sum_k \left(e^k\right)^2$$

where the output errors are, as before

$$e^k = o^k - d^k$$

and the sum is taken over all input patterns. The total error E is a function of the weights, and the learning problem can be expressed as trying to find the values of the weights that minimise E. The minimisation can be performed, for example, through gradient descent. The partial derivative of E relative to a weight can be easily seen to be

$$\frac{\partial E}{\partial a_i} = 2\sum_k x_i^k e^k$$

where x_i^k is the i-th component of pattern $\boldsymbol{x}^k$. Therefore, the gradient minimisation procedure corresponds to updating the weights according to

$$a_i^{(n+1)} = a_i^{(n)} - \eta \sum_k x_i^k e^k$$

where η is a suitable, positive learning rate parameter. One can show that E, as a function of the weights, is of second degree and concave upwards, and therefore has a single minimum. This guarantees that the procedure will converge to that minimum, for sufficiently small values of η.

To apply this procedure, one must sweep through all patterns before each weight update, to be able to compute all products $x_i^k e^k$. This is rather inconvenient when the number of patterns is large, and Widrow and Hoff devised a faster method, which simply updates the weights after the presentation of each pattern $\boldsymbol{x}^k$, according to

$$a_i^{(n+1)} = a_i^{(n)} - \eta x_i^k e^k$$

This faster procedure is what is known as the *delta rule*, or *LMS rule* (for Least Mean Squares). It is not a pure gradient minimisation. However, it is easy to see that if the parameter η is sufficiently small, the net effect after each sweep through all patterns is

very similar to what would be obtained with the exact gradient procedure. Under certain conditions, this rule corresponds to what is called a *stochastic gradient descent*, and its convergence can still be guaranteed.

The adaline is a linear unit, and therefore is very limited in its capabilities. However, linear operations form the basis of many signal processing techniques, and the LMS rule, or variations upon it, are often used, for example, in various kinds of adaptive filters, like echo cancellers and adaptive equalisers.

Multilayer feedforward perceptrons

The perceptron, as introduced by Rosenblatt, is often called a *single layer system*, since its single unit cannot, of course, form more than one layer (of units)[2]. A multilayer perceptron would be formed by two or more units, with the outputs of one or more of them being used as inputs to other units (see figure 4). The limitation to hyperplane boundaries comes from considering only single layer systems: a two layer system can form any convex boundary, and systems with three or more layers can form any desired boundary shapes, even concave and/or formed by several disconnected parts (provided, in both cases, that enough units exist in the intermediate layers)[3]. Figure 5 gives an example of a two layer perceptron implementing the XOR function.

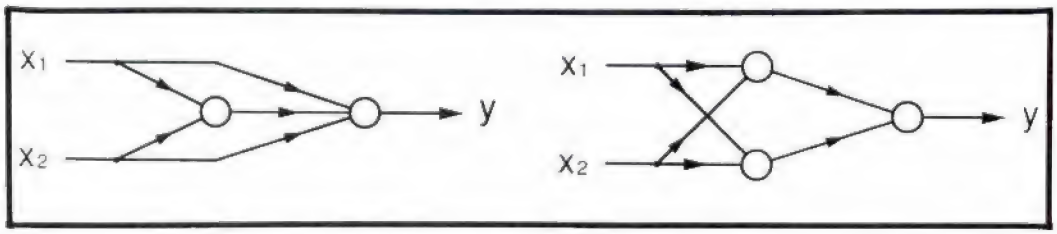


figure 4 Two simple examples of multilayer perceptrons. Units are represented as circles. Each unit has weights in each of its input branches, plus a bias term.

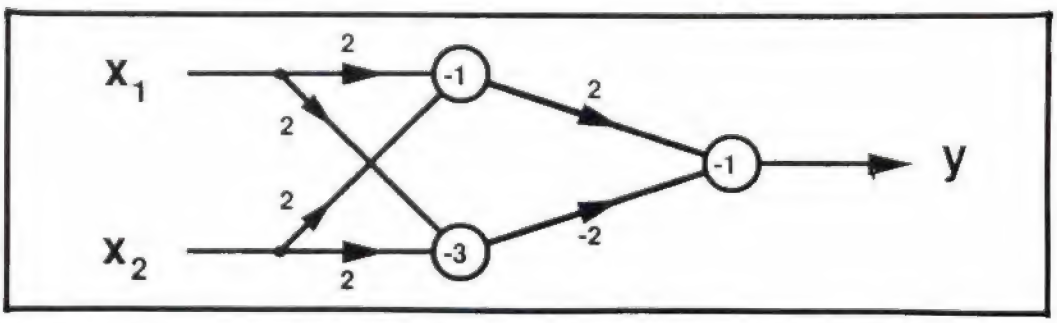


figure 5 A two layer perceptron implementing the XOR function. Each unit is represented by a circle. In this figure, as well as in figure 8 and 13, the value of the bias term a_0 for each unit, is indicated inside the corresponding circle. The values of the other weights are indicated beside the respective branches.

Learning in multilayer systems raised a hard problem. Consider the left hand network in figure 4, and assume that for a given input pattern, the output produced by the perceptron was wrong. Which weights should one modify, in order to attempt to correct this situation? Those of the output unit? Or those of one (or more) of the units in the

2 Some authors also refer to inputs as units. We do not use that nomenclature here.

3 Note, however, that this kind of reasoning does not apply to adalines: a system with multiple layers of linear units is always equivalent to a single linear unit.

intermediate layer? If so, which unit(s)? and by how much? This difficulty came to be known as the *credit assignment problem*: when there are the units that are not directly connected to the outputs[4], which of these units should be blamed for an erroneous system output?

In 1974, Paul Werbos introduced a learning rule for multilayer perceptron-like systems, which later came to be called *backpropagation*, or *generalised delta rule* [4], effectively giving a solution to the credit assignment problem. This work stayed largely unknown, and the same rule was reinvented, in the middle eighties, by Parker and Le Cun. The widespread knowledge of this rule came from a publication by Rumelhart, Hinton and Williams [5]. As we said above, they considered perceptron-like systems. The changes made to Rosenblatt's perceptron were:

1 - They considered multilayer systems with any number of units, and any pattern of interconnections between the units, restricted only to being *feedforward*, i.e., the interconnections between units cannot form any loops.

2 - The step function in Rosenblatt's units was smoothed. One of the most commonly used "smoothed step functions" is

$$y = S(s) = \frac{1}{1+e^{-s}}$$

which is sketched in figure 6. Functions with this general shape are often called *sigmoids*[5].

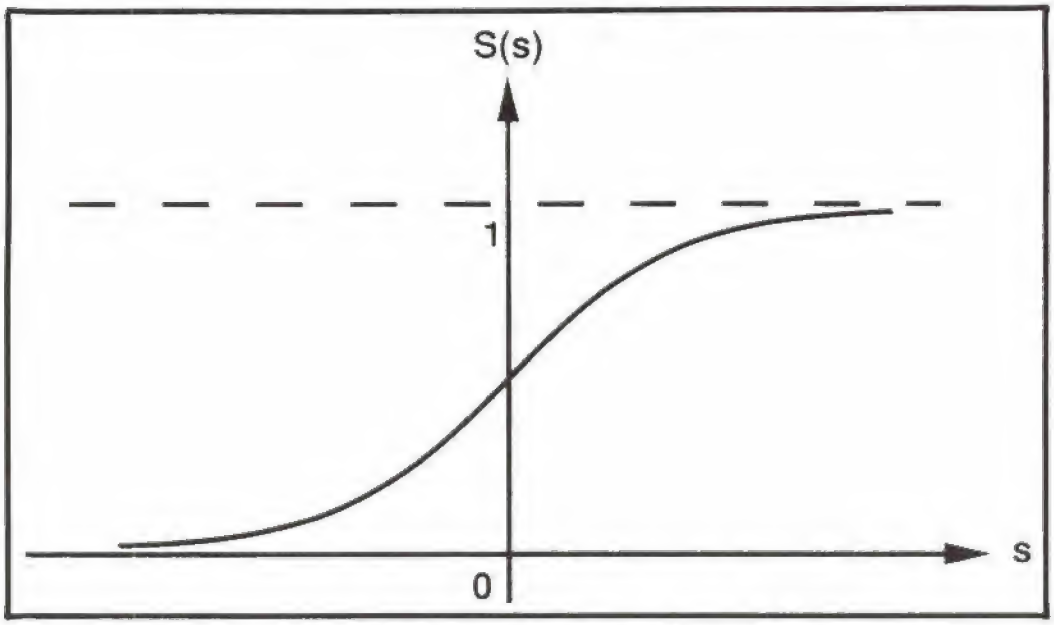


figure 6 Sketch of a sigmoid.

We shall call these systems *feedforward multilayer perceptrons*. They can be used for computing binary valued functions, but they can be used for computing analog valued ones just as well, since their units are analog.

---

4 These units are often called *hidden units*.

5 The backpropagation algorithm can however be used with any continuous, differentiable function *S*, even nonmomotonic. The function *S* may also differ from unit to unit.

As in the delta rule, we can measure the network's performance through the total quadratic output error. But since we have smoothed the step function, E is a differentiable function of the weights of the various units in the network, and we can minimise it through gradient descent. The resulting procedure is easier to understand by means of an example. Let us consider the network of figure 7a.

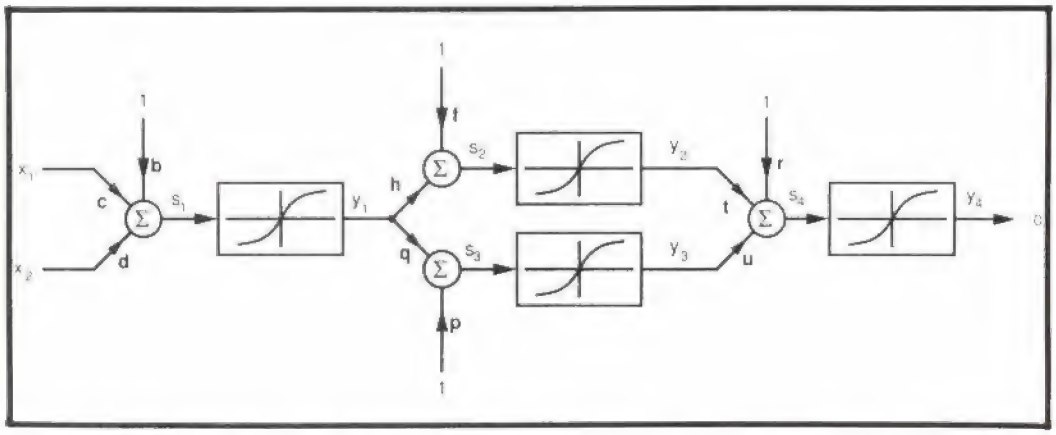


figure 7a Example of a feedforward perceptron network. Bold characters indicate weights, light characters indicate network variables. The weights have been denoted by non-indexed letters for simplicity.

We can apply a specific pattern at the input, and compute the values that will be obtained at the various nodes, as indicated in the figure. If we *linearize* the network, each sigmoid box will be replaced by a linear branch with a gain $g_i = S'(s_i)$. We can then apply to the network, an operation called *transposition* [6], which corresponds to reversing the direction of flow of all branches, keeping their gains unchanged. Of course, outputs will now become inputs, and vice-versa. Also, points where branches diverged in the original network will now be replaced by summations, and points of summation will be replaced by divergences. The result of applying these two successive operations to the network of figure 7a is shown in figure 7b. This will be called the *error propagation network*, for reasons that will become apparent ahead.

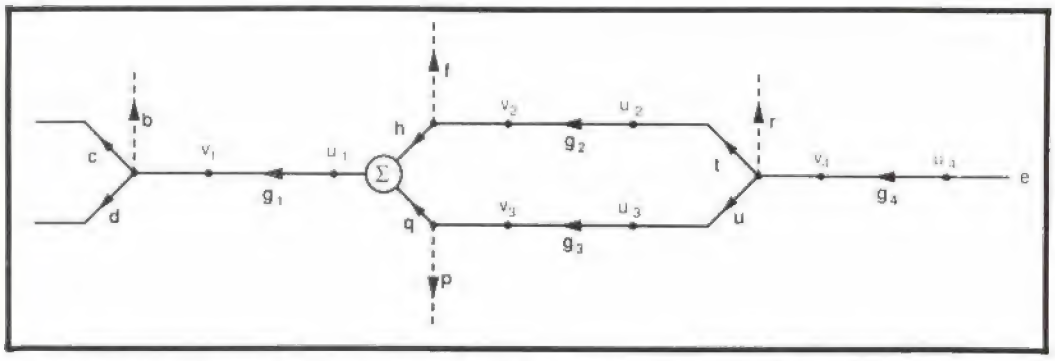


figure 7b The error propagation network, obtained by linearizing and transposing the network of figure 7a. Dashed parts are not needed for applying backpropagation.

Assume that we apply the output error of the perceptron network of figure 7a to the input of the error propagation network, as shown in figure 7b. Through repeated application of chain differentiation, it is easy to conclude that, for the given input pattern, the partial derivative of the squared output error e^2 relative to any desired weight is given

by a simple rule: it is twice the input of that weight's branch in the perceptron network, multiplied by the input of the same branch in the error propagation network[6]. For example,

$$\frac{\partial e^2}{\partial b} = 2 \cdot 1 \cdot v_1$$

$$\frac{\partial e^2}{\partial c} = 2 \cdot x_1 \cdot v_1$$

$$\frac{\partial e^2}{\partial u} = 2 \cdot y_3 \cdot v_4$$

where x_1 and y_3 are taken from the perceptron network (figure 7a), and v_1 and v_4 are taken from the error propagation network (figure 7b).

Of course, the partial derivates of the total error E can be obtained by adding up each of these derivates for all input patterns. After these derivatives have been computed, the weights can be updated. For example, the update of weight u would be performed according to

$$u^{(n+1)} = u^{(n)} - \eta \sum_k y_3^k v_4^k$$

where the upper index k denotes, as before, the values obtained on the networks when the pattern x^k is presented at the input of the perceptron.

This is the exact gradient procedure. The fact that it involves a backward propagation of errors justifies the name by which it is most widely known, *backpropagation*. As in the case of the delta rule, a variant of this method consists of updating the weights immediately after each pattern presentation. For example, for weight u, after the presentation of x^k

$$u^{(n+1)} = u^{(n)} - \eta y_3^k v_4^k$$

This latter procedure is sometimes called *real-time* backpropagation, and correspondingly the former, exact gradient one is called *batch* backpropagation. As with the delta rule, real-time backpropagation can, under certain conditions, be shown to be a *stochastic gradient procedure*, which is then guaranteed to find a local minimum of the total error E. Backpropagation can be considered a generalisation of Rosenblatt's learning rule, as we saw, but it can also be viewed as a generalisation of the delta rule. In fact, if we consider a single unit, with $S(s)=s$, the real-time version of backpropagation is exactly the delta rule of Widrow and Hoff. That is why backpropagation is also called *generalised delta rule*.

Reminding ourselves that the gains g_i in the error propagation network are given by $g_i=S'(s_i)$, we see that the weights of the output unit can be written in the following form (we give an example for weight u only):

$$u^{(n+1)} = u^{(n)} - \eta S'(s_4) y_3 e$$

6 This result is relatively easy to obtain, but the derivation will not be given here, since this is a special case of a more general result, to be derived in the next section.

Comparing this equation with Rosenblatt's learning rule expressed in equation (1), and noting that y_3 plays here the role of x_i in that equation, i.e. an input to the output unit, we can see that what we have added in backpropagation is the factor $S'(s_i)$. This factor, used in each of the units, plus the use of weights for propagating the error backwards in interconnection branches, makes each unit "see" the output error with a different magnitude (and possibly a different sign). It is by this weighting of the output error as seen by each of the units, that backpropagation effectively solves the credit assignment problem: the "blame" for the output error is distributed among units, with appropriate magnitudes and signs.

Gradient based optimisation techniques have difficulties in dealing with ravines. The problem is best visualised by assuming that E is a function of just two variables, that is represented by a surface in three dimensional space. Imagine that this surface has a narrow ravine, whose bottom has a gentle slope, and assume that we start our optimisation somewhere on one of the flanks of this ravine. Since the flanks have a very steep slope, the step size parameter η will have to be made very small, to prevent a strong, divergent oscillation between the two flanks. If η is made small, we will be able to go down to the bottom of the ravine, but then we will proceed at a very low speed, since the slope at the bottom is very small. Increasing η will again result in oscillation, since we are never exactly at the bottom. This problem may be alleviated by use of a *momentum term* [5]. The central idea is to make steps much larger in directions in which there is no oscillation, and to attenuate them in directions in which there is oscillation. This is done by effectively performing a recursive low-pass filtering on weight updates. Using weight u again as an example,

$$\Delta u^{(n+1)} = \alpha \Delta u^{(n)} - \eta \sum_k y_3^k v_4^k$$

and

$$u^{(n+1)} = u^{(n)} + \Delta u^{(n+1)}$$

This low-pass filtering can be seen as progressively giving momentum to the optimisation along the ravine's direction, while effectively attenuating oscillations. However, it may also cause an undesired effect: if there is a bend somewhere in the ravine, too much momentum may have been acquired along its previous direction, and we may not be able to turn fast enough. We will then find ourselves climbing one of the flanks, and perhaps even losing the ravine completely. To be able to turn faster, the momentum parameter α must be decreased. Hand tuning of the optimisation parameters η and α during the whole optimisation process, together with a close monitoring of the error E, retracing the last step whenever it increases, can mean the difference between success and failure, in hard problems.

Backpropagation extends quite easily to systems with multiple outputs. The performance measure is now the sum of the squared errors over all input patterns and all outputs. The error propagation network is obtained just as before, the output errors are applied all at the same time to the corresponding inputs of the error propagation network, and the gradient components are computed using the same expressions as for single-output systems. Backpropagation is also easily extendable to non-quadratic measures of performance, as long as they are differentiable, implying only a modification in the values that are input to the error propagation network.

One can readily see from the weight update equations that if a hardware implementation is desired, the perceptron and the error propagation network can be placed "side

by side", with corresponding branches physically close to each other, and then computation of weight updates requires only local connections. This *local property* is very important when networks with large numbers of units are envisaged, because non-local connections in huge numbers would certainly constitute one of the major difficulties of implementation. A very appealing property of hardware implementations of perceptrons (and of most neutral network architectures) is that in the operation of the network, and also in its training, all units can operate in parallel. This has been called *massive parallelism*, since the number of operations running in parallel can be enormously large, and it means that such implementations can potentially achieve speedups of several orders of magnitude relative to more conventional approaches.

Readers familiar with optimisation techniques may have a doubt by now. Backpropagation, being based on steepest descent, is guaranteed to find a local minimum of E (albeit in infinite time), but may never find the global minimum. Will this impair the performance of these networks? The answer is a qualified yes. One would certainly prefer to have a usable global optimisation procedure, but local minima have shown to be a relatively minor difficulty. In some simple cases, it has been experimentally shown that adding one or a few extra hidden units will allow the system not to fall into such minima. In more complex cases, it is virtually impossible to know whether the system is at a local or global minimum, and most often one should realistically assume that only local minima are found. However, experiments such as those reported below have shown that even these local minima often are able to solve the given problems with adequate accuracy.

Two examples of multilayer perceptrons trained by backpropagation to perform the XOR function are shown in figure 8. Examples of situations where multilayer perceptrons have been used to deal with more complex problems are appearing steadily. One of the best known is NETtalk [7], a multilayer perceptron trained to perform text-to-phoneme transcription in English. The best result reported for this system is a performance of 97.5% correct individual phonemes, on a corpus of about 20,000 words. Other applications of multilayer perceptrons that have been reported include recognition and coding of speech and images, recognition of hand-written characters, evaluation of loan applications, and even driving a vehicle in real-time on a road. In these examples, the training of the perceptron and its operation have generally been simulated on digital computers. This often posed limitations to the size of the systems that could be considered, since simulating highly parallel systems in sequential machines is a very inefficient process.

One of the very interesting capabilities of perceptrons, which is common to other neural network structures, is *generalisation*: the ability to detect the regularities present in the training data, and to use them to yield appropriate outputs for input patterns that were never presented during training. As an example NETtalk, when trained on the 1000

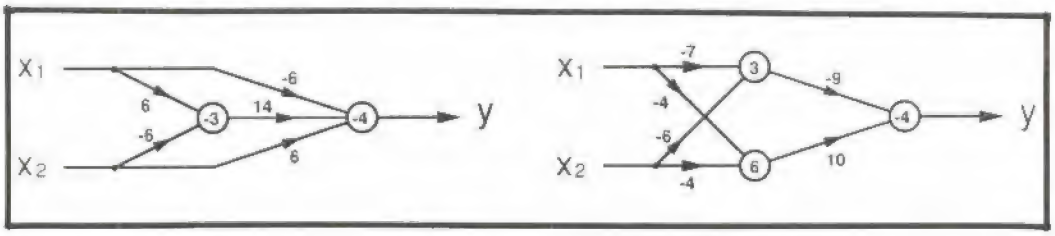


figure 8 Examples of two perceptrons taught by backpropagation to implement the XOR function. The output is considered as logical 1 whenever the value of y is above .5, and logical 0 when it is below .5. Weight values have been rounded, for convenience.

most frequent English words only, yielded 80% correct results on the 19,000 words that it had never seen during training. While this result is far from perfect, it is much above chance, which would correspond to less than 10% correct. This shows that the network was able to infer transcription "rules" (or better, regularities) from the 1000 training words, which it then used to transcribe the new words. Generalisation properties of neural networks depend on a number of factors, like training set size, and network size and topology, and cannot yet be considered well understood.

As a final remark, we should note that plain backpropagation, as described here, even with the use of the momentum term, is not the fastest method available for training multilayer perceptrons. Faster versions of backpropagation range from simple ones involving the use of multiple, adaptive step size parameters, to more elaborate ones, using second-order derivatives or *conjugate gradient* techniques.

Recurrent networks

An important restriction of the perceptron structures discussed above is that they must be feedforward, i.e., that they cannot contain loops. Although there is no fundamental limitation to the functions that feedforward structures can implement, recurrent networks, i.e. networks in which connections between units can form loops, can be more efficient than feedforward ones in some cases (e.g., by implementing the same function with a smaller number of units). Advantageous use of recurrent networks has been reported in the areas of speech recognition and image processing, for example. Furthermore, recurrent networks can be used in other operating modes, which are not available in feedforward systems, as shown in the examples below.

A recurrent network has the same structure as a feedforward one, except that now there is no restriction to the interconnections between units, these interconnections being allowed to form loops. While the way to operate feedforward networks is quite straightforward, several different possibilities arise when feedback is allowed. It is therefore important to clarify how we are assuming that the networks are to be operated. The network operates in continuous time (as opposed to discrete time steps). When a given, fixed pattern is presented at its input, a recurrent network can exhibit different kinds of behaviour: it can have one or more equilibrium states (each of which can be stable or unstable), it can oscillate, or can even exhibit chaotic behaviour. We will assume that the networks we are considering do not exhibit oscillatory or chaotic behaviour[7]. We present each pattern long enough for the network to stabilise. Network outputs are observed, and compared with the desired outputs, only after stability is reached. A system with feedback can also exhibit sequential behaviour, i.e., its present outputs can depend not only on the present input pattern, but also on past ones. For the time being, we will consider only nonsequential desired outputs, i.e., outputs that depend only on the present inputs.

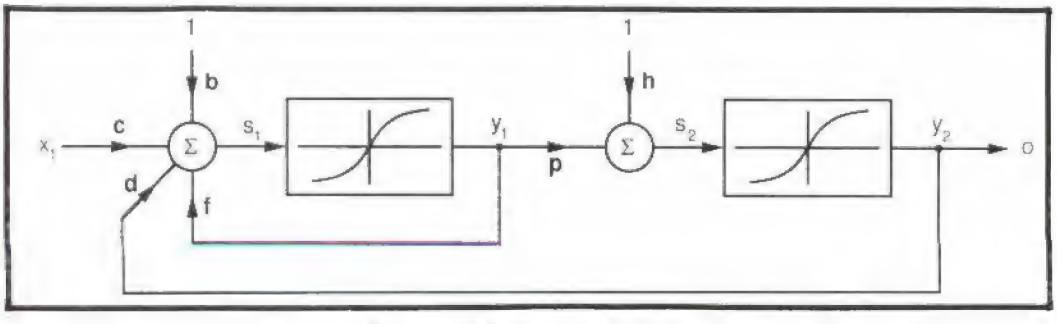


figure 9a A simple recurrent network.

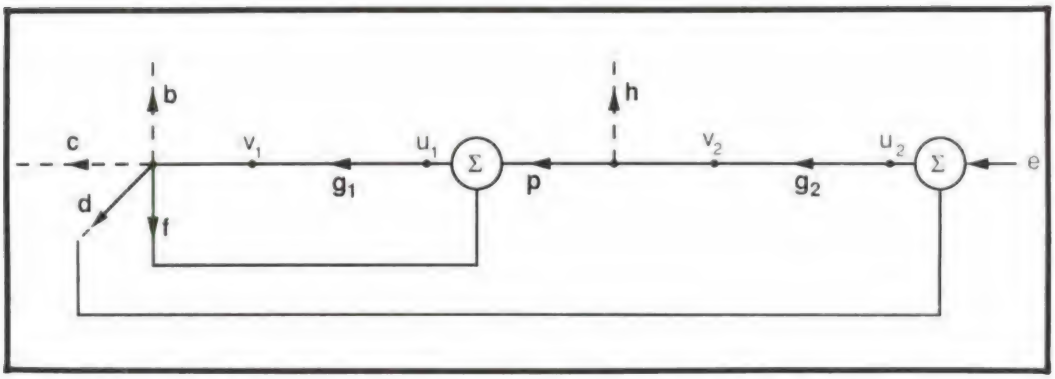


figure 9b The error propagation network, obtained by linearization and transposition of the network of figure 9a. As in figure 7b, dashed parts are not needed for applying backpropagation.

One can wonder if backpropagation, as described so far, can be generalised to these systems. The extension of backpropagation to recurrent networks was first derived by this author [9]. Shortly afterwards, Pineda independently derived the same result [10]. The extension has a very natural form. As we shall show below, a backward error propagation procedure can still be used for computing the gradient of the quadratic error relative to the weights. The error propagation network is still derived by linearization and transposition of the perceptron network (see figure 9 for an example). Since the perceptron network is now recurrent, the error propagation network has loops, corresponding to those that exist in the perceptron itself. The existence of loops in the error propagation network raises no special problems, except that, after applying the errors to this network, one must let it stabilise before observing its node values for the computation of partial derivatives. The learning procedure is then performed as follows:

1) Apply a pattern to the inputs of the perceptron, and let that pattern remain fixed, until step 4, below.

2) Let the perceptron reach equilibrium, and then observe its outputs.

3) Compute the output errors, and apply them to the inputs of the error propagation network.

4) Let the error propagation network stabilise.

5) Compute the gradient components by means of the values taken from the perceptron and the error propagation network, exactly as in the feedforward case. For example, referring to figure 9,

$$\frac{\partial e^2}{\partial c} = 2 \cdot x_1 \cdot v_1$$

$$\frac{\partial e^2}{\partial d} = 2 \cdot y_2 \cdot v_1$$

As in the feedforward case, this procedure must be repeated for all input patterns. As in that case, the weights can be updated after the presentation of each pattern (for real-time training) or after each sweep through all patterns (batch training). One might be concerned about the stability of the error propagation network, but this network has

7 A short discussion on stability is made ahead. This subject, as well as the one of multiple stable states are considered at greater length in [9].

been shown to be stable if its dynamics are matched to those of the perceptron network [9].

To derive the gradient learning procedure, consider first a non-linear network ***P*** (possibly recurrent) with fixed external inputs x_i, an output *o* and a linear branch of gain *a*, as in figure 10a (a single input *x* is shown in the figure, for simplicity). We shall assume that the network is at an equilibrium point, and the same assumption will be made for all networks that follow, in this derivation.

Now assume that we give an infinitesimal increment to the gain *a*. This can be accomplished by adding an extra branch with gain *da* (figure 10b). This will cause increments in all node variables (including node *y*, at the input to the branch under consideration, since there can be feedback connections in the network). The output of the new branch is *(y+dy).da*, which we can simplify to *y.da*, since the higher order term *dy.da* can be discarded. The same result can be obtained using an external input with value *y*, through a branch with gain *da* (figure 10c).

Now linearize the network around the original equilibrium point, considering only increments in the variables. The result will be network ***PL***, of figure 10d. The original external input(s) *x* will not be present in the linearized network, since they have suffered no increment. We can obtain the value of $\partial o/\partial a$ by dividing *do*, the output of this network, by *da*. This can be accomplished by dividing the network's input by *da*, since the network is linear. We shall do this by changing the gain of the input branch to unity (figure 10e). The output, $\partial o/\partial a$, shall be designated by $\dot{o}$, for compactness.

Assume that we now transpose this network, obtaining network ***PLT*** (figure 10f). The transposition theorem [6] states that, when we transpose a linear network with a single

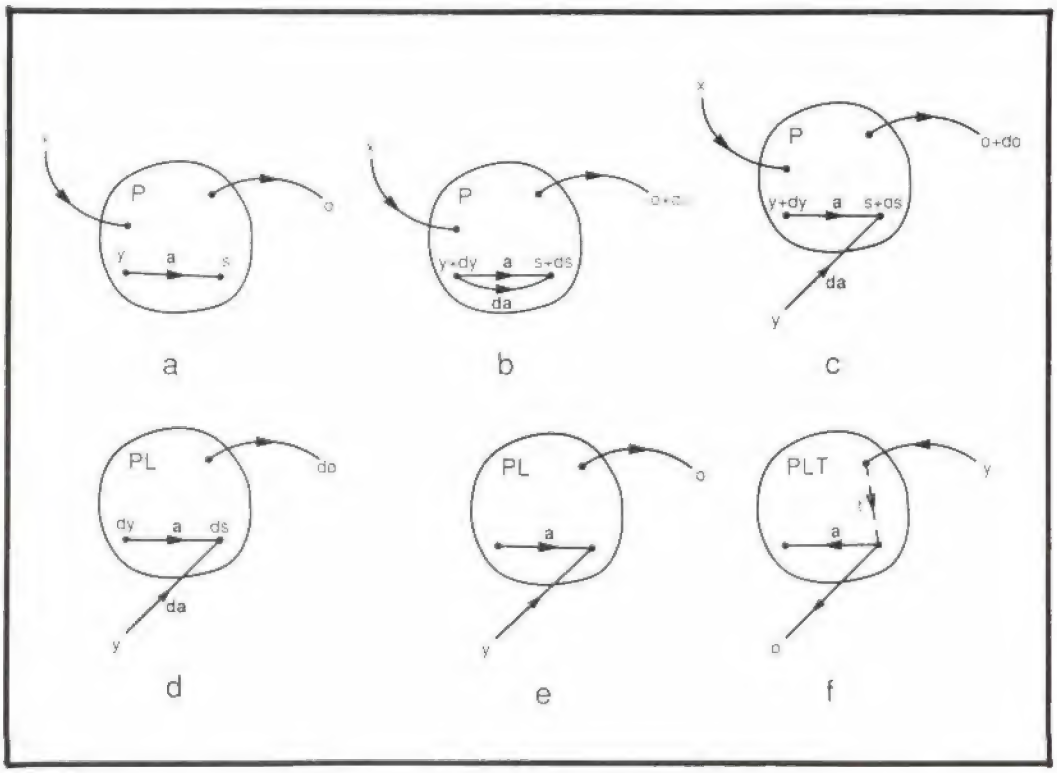


figure 10 Computation of the partial derivative of the output of a non-linear network, relative to a branch weight. See text for explanation.

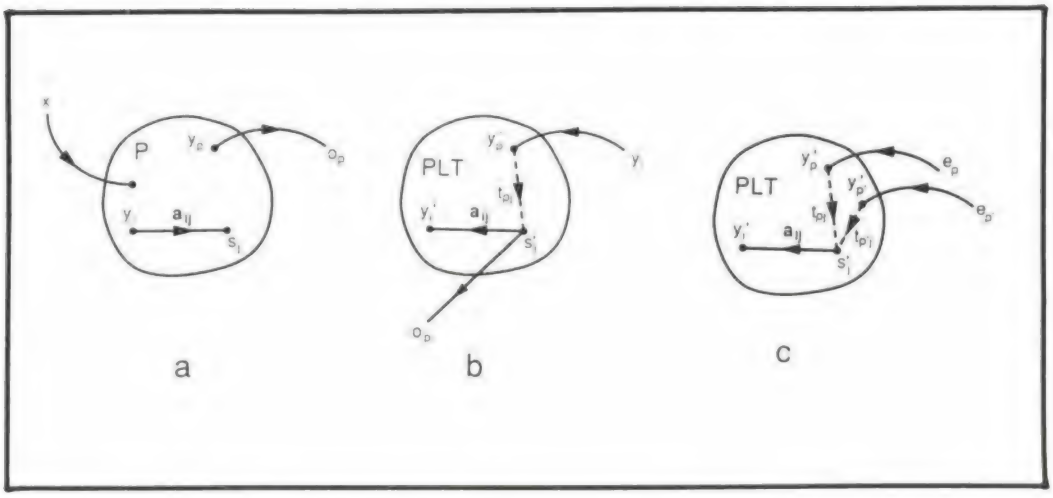


figure 11 Computation of the partial derivative in the case of a recurrent network. See text for explanation.

input and a single output, we obtain a network with the same input-to-output relationship. This means that, when the input to network ***PLT*** is y, the output will still be $\dot{o}$, as shown in the figure. Designating by t the gain from input to output in this network, we can write

$$\dot{o} = yt$$

Having derived an expression of the derivative of an output relative to a branch gain, let us now consider a recurrent perceptron with several outputs o_p, as in figure 11a (once again, a single input is drawn, for simplicity). Node y_i is the output of some unit i, s_j is the input sum of some other unit j, and a_{ij} is the weight connecting unit i to unit j. Some fixed pattern is assumed to be present at the input. For each output, we can write (figure 11b)

$$\dot{o}_p = \frac{\partial o_p}{\partial a_{ij}} = y_{ij} t_{pj} \qquad (15)$$

Designate by O the set of indexes of the units that produce external outputs. The squared error for the given input pattern at the output o_p is

$$e_p^2 = \left(o_p - d_p\right)^2$$

d_p being the desired value of that output for the input pattern we are considering. Therefore,

$$\frac{\partial e^2}{\partial a_{ij}} = 2 \sum_{p \in O} e_p \dot{o}_p$$

and using equation (2) above

$$\frac{\partial e^2}{\partial a_{ij}} = 2 \sum_{p \in O} e_p y_i t_{pj}$$

or, since y_i does not depend on the summation index,

$$\frac{\partial e^2}{\partial a_{ij}} = 2 y_i \sum_{p \in O} e_p t_{pj}$$

Examining figure 11c, and remembering that network ***PLT*** is linear, we finally arrive at

$$\frac{\partial e^2}{\partial a_{ij}} = 2 y_i s'_i$$

in which s'_j is the value observed at the respective node when the errors are all simultaneously applied to the corresponding inputs of network ***PLT*** (the error propagation network). This result extends the backpropagation rule of feedforward networks in a natural way: the derivative relative to a weight is still twice the product of the inputs of the corresponding branch in the perceptron and in the error propagation network. The error propagation network is still obtained through linearization and transposition of the original network.

This justifies that training of these networks be performed in the way that we described above: a pattern is presented at the input, and the network is allowed to stabilise. The network's outputs are then compared with the desired ones, yielding the output errors, which are input to the error propagation network. This network is allowed to stabilise, and afterwards its node values s'_j, together with the unit activations of the recurrent perceptron, y_i, are used to compute weight updates, according to

$$\Delta a_{ij} = -\eta y_i s'_j$$

As in the feedforward case, weights can be updated immediately, or updates can be accumulated for all training patterns, depending on the version of backpropagation that we want to implement.

Until now, we have assumed that the recurrent network is stable for any input pattern. This stability issue deserves some further attention. First of all, we should note that while the locations of the equilibrium states of the network depend only on the values of the interconnection weights, the stability of these states depends also on the dynamical behaviour of the units of the network. Commonly assumed dynamical properties of units correspond to the circuits shown in figure 12. These circuits come from considerations on the dynamical behaviour of actual neurons, and also from considerations on the properties of electronic circuits that could be used to implement the units. For both kinds of dynamical behaviours, it has been shown [9,11] that a sufficient condition for the stability of the network can be guaranteed by forcing the weights to be symmetrical, a condition that is easy to incorporate into the backpropagation algorithm: one just has to initialise the network with symmetrical weights and, before each weight update, one must take the sum of Δa_{ij} and Δa_{dg}, computed as before, and update both a_{ij} and a_{dg} by this sum. Experimental results reported below, suggest that weight symmetry may not significantly restrict the capabilities of recurrent networks.

Incorporation of backpropagation learning in an analog implementation of a recurrent network is not expected to raise any extra problems, relative to the implementation of feedforward networks, since a stable analog network running in continuous time will find equilibrium by itself. However, digital hardware implementations, as well as computer simulations, must incorporate a special provision (usually an iteration) for finding stable equilibrium. From the author's experience, this means an increase in computation by a factor of roughly 3-6, relative to feedforward networks with the same number of weights.

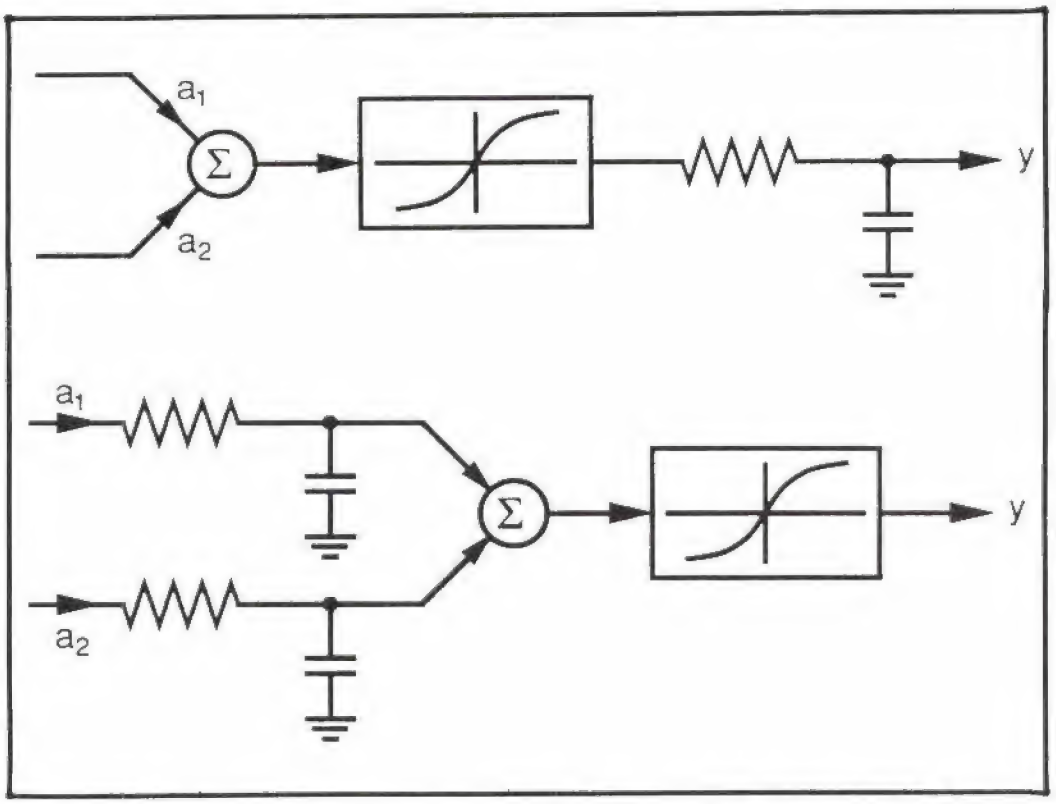


figure 12 Two circuits corresponding to dynamical behaviours often assumed for network units. Resistor and capacitor values may differ from unit to unit, or from branch to branch.

Experience with recurrent networks is small[8]. The author has performed some tests on relatively simple problems, and has found that the advantage of feedback can be quite significant in some cases [9]. As an example, consider a problem in which there are 3 analog inputs, each of them with a value randomly selected between -1 and 1. There are 3 outputs, and we wish them to indicate which of the inputs is largest. The output corresponding to the largest input should be above .5 (interpreted as logical 1), and the other two outputs should be below .5 (interpreted as logical 0). Table I shows the results obtained on feedforward perceptrons of various sizes, and on a 3-unit recurrent network, in this problem[9].The recurrent network clearly outperforms even a 6-unit feedforward one, while employing a much lower number of weights. It does so by developing strong negative weights from each unit to the other two (this is often known as *lateral inhibition*). With these weights, the unit receiving the strongest activation from the inputs tends to shut the other two almost completely off. These units being shut off, they will not prevent the other one from reaching a strong activation, and producing the correct result.

---

8 There have been reports of advantages of non-sequential recurrents networks, like those we have been considering, over feedforward ones, in the areas of speech recognition and image processing. However, the author does not yet know of any such report available in printed form.

9 All networks considered both in this example and in the following one are fully connected, i.e., there are weights a_{ij} for all i and j, except for the restriction to feedforward structures where indicated. Since there is still little knowledge as to the best topologies to use in recurrent networks, this option was chosen to ensure a fair comparison between feedforward and recurrent structures.

Table 1
Results for the "largest of 3 inputs" problem.

Structure	Number of independent weights	Average error rate, per output
3 units, feedforward	15	5.9%
4 units, feedforward	22	5.5%
5 units, feedforward	30	3.5%
6 units, feedforward	39	3.6%
3 units recurrent	21	1.7%
3 units recurrent, symmetrical	18	1.4%

In another example, a *pattern completion* problem was considered. Ten vectors were randomly generated, each with ten binary components. At each pattern presentation, only eight of these components were shown to the network, which was required to complete the other two. The components that were to be guessed were randomly selected, at each pattern presentation. Therefore, the same components would sometimes act as inputs, and other times as outputs.

Recurrent networks are more naturally suited to this kind of problem, since the known components can be directly clamped on units (i.e. each unit's output can be forced to take the value of the corresponding, known component), while the units corresponding to unknown components are left free, and their outputs are observed after reaching a stable state. Feedforward networks must have well defined inputs and outputs, and therefore are much harder to adapt to this kind of problem. Table II depicts the results obtained, and shows the clear superiority of recurrent networks in this kind of problem.

Table 2
Results of pattern completion problem.

Structure	Number of independent weights	Average error rate, per missing component
10 units, feedforward	155	4.50%
10 units, recurrent	110	0.89%
10 units, recurrent, symmetrical	65	0.86%

While recurrent networks have shown a definite advantage in these two cases, one should not conclude the same will happen in every problem. Clarification of the kinds of situations in which recurrent structures are desirable is a task still to be performed.

Recurrent perceptrons in sequential applications

Recurrent networks can also be used in sequential applications, i.e. in those applications in which the desired output depends not only on present inputs, but also on past ones. Recently, several algorithms have been introduced in the literature, for training the time-dependent behaviour of recurrent networks, both in time-continuous and time-discrete modes of operation. However, most of these algorithms cannot be viewed as simple extensions of backpropagation, and are too complex to be described here. In what follows, we will only be concerned with the simplest and earliest extension of backpropagation to time-discrete sequential applications.

To be used in a time-discrete sequential manner, recurrent networks must be operated in a mode which is different from the one we have considered in the last section. Successive input patterns are presented to the network at successive, discrete time intervals. An analog memory (e.g. a sample-and-hold element) is incorporated at the output of each unit. In each discrete time step, after a pattern is presented at the input,

all memory elements are triggered synchronously. Then, at the next time step, a new pattern is presented and a new trigger is applied, and so on. It is easy to see, as was already suggested by Minsky and Papert [3] and later shown by Rumelhart, Hinton and Williams [5], that such a network can be conceptually "unfolded" in time, by opening the interconnections at the memory elements, and repeating the network once for each time step being considered. This gives rise to an equivalent nonsequential feedforward network that can be trained in the usual way by backpropagation (see figure 13).

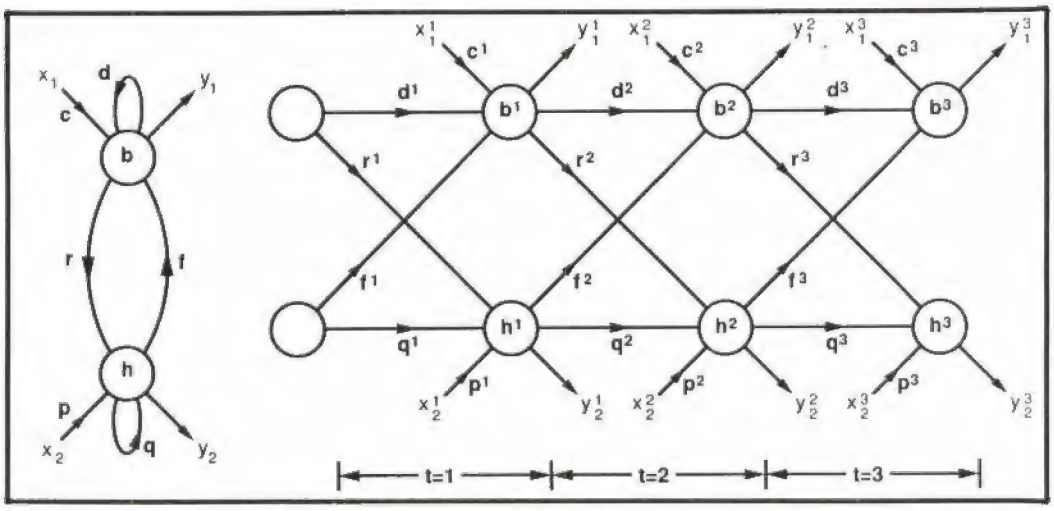


figure 13 Example of a sequential perceptron (left) and of its unfolding for three time steps (right). Upper indexes in variables indicate the time steps that those variables correspond to. The leftmost, open circles in the unfolded network represent the initial states of the units.

In this unfolded network, the various occurrences of each weight correspond to the same single weight value in the recurrent network, and must therefore have the same value. It is easy to show, by computing partial derivatives subject to this restriction, that the updates corresponding to those various occurrences must all be added together, to compute the update to the single weight of the sequential network. For example in figure 13, after computing the updates for the weights b^1, b^2 and b^3 in the normal way, the update of weight b in the recurrent network is computed according to

$$\Delta b = \Delta b^1 + \Delta b^2 + \Delta b^3$$

and the weights b^i of the unfolded network are then simply set equal to the updated value of b.

This training procedure is often called *unfolding in time* or *backpropagation through time*. Its extension to networks in which only some units have memory elements at their outputs is easy. As before, one must open interconnections at the memory elements, and then unfold the network in time, as we did above. Loops which contain at least one memory element will be opened, becoming feedforward. However, loops without any memory element in them will be kept in the unfolded network, which will then be recurrent, requiring the recurrent version of backpropagation studied above.

As a simple example of the use of backpropagation in sequential perceptrons, a network can be taught to operate as a shift register [5]. In a more practically oriented application, sequential networks have already been employed in speech recognition experiments, in which parameters from successive speech frames were presented at the input in successive time steps, instead of being presented all at once to a much larger nonsequential system [12].

The reader should note that the amount of training needed by sequential networks tends to grow very fast with the input sequence length, because the number of units in the unfolded network grows linearly with this length. Furthermore, implementation of the learning procedure requires the existence of as many memory elements in each unit as there are time steps in the training sequence, because training can only be performed at the end of the sequence, but needs all intermediate values. This tends to make hardware implementation difficult for reasonably long input sequences.

Conclusion

Backpropagation is a powerful supervised learning rule for neural networks, both feed-forward and recurrent. It has found widespread use in the training of systems for a wide range of applications. However, one should realise that it still presents some significant limitations. One of them is that it tells us nothing about the desirable size and topology of a network, for dealing with a specific problem. Some work has already been done on the dynamical modification of network topology. However, one cannot yet say that such a problem is adequately solved. Such work is beyond the scope of the present paper.

Another important limitation is speed. Backpropagation is certainly one of the fastest learning rules known today, which can be applied to many useful problems. However, its use in a large problem (noise reduction in speech signals, for example) may involve anything between, say, one day and one month of CPU time on a workstation, which makes this kind of problem one of the largest that can presently be handled. Of course, massively parallel implementation of networks will in the future yield performance increases of several orders of magnitude, allowing us to deal with larger problems. However, backpropagation, like most neural network learning algorithms, tends to scale badly with problem size, and therefore a large increase in speed may only bring about a relatively modest increase in the size of problems that can be handled. Some fundamental improvements in learning algorithms may still be needed before many important problems are within our reach.

References

[1] Rosenblatt, F. (1962). *Principles of Neurodynamics*, Spartan, New York.

[2] Widrow, B., Hoff, M. (1960). Adaptive switching circuits, *IRE WESCON Convention Record.*

[3] Minsky, M., Papert, S. (1969). *Perceptrons*, Cambridge, MA: MIT Press.

[4] Werbos, P. (1984). *Beyond Regression: New Tools for Prediction and Analysis in Behavioral Sciences*, PhD thesis in applied mathematics, Harvard University.

[5] Rumelhart, D., McClelland, J. and the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, **1**, Cambridge, MA: MIT Press.

[6] Oppenheim, A., Schafer, R. (1975). *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall.

[7] Sejnowski, T., Rosenberg, C. (1987). *Parallel Networks that Learn to Pronounce English Text*, Complex Systems **1** (**1**).

[8] Pomerleau, D. (1989). ALVINN: an Autonomous Land Vehicle in a Neural Network, in D.S. Touretzky (ed.), *Advances in Neural Information Processing Systems I*, San Mateo, CA: Morgan Kaufmann Publishers.

[9] Almeida, L. (1987). A Learning Rule for Asynchronous Perceptrons with Feedback in a Combinatorial Environment, *Proceedings of the IEEE First Annual International Conference on Neural Networks*, S.Diego, CA. Also Backpropagation in Non-feedforward Networks (1989) in I. Aleksander (ed.), *Neural Computing Architectures*, London: North Oxford Academic Publishers .

[10] Pineda, F. (1988). Generalization of Backpropagation to recurrent and Higher Order Neural Networks, in D. Anderson (ed.), *Neural Information Processing Systems*, NY: American Institute of Physics, New York.

[11] Hopfield, J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons, *Proceedings of the National Academy of Sciences of the USA*, **81**, 3088-3092.

[12] Watrous, R. (1987). Learning Phonetic Features using Connectionist Networks: An Experiment in Speech Recognition, *Proceedings of the IEEE First Annual International Conference on Neural Networks*, S. Diego, CA.

Sobre o Autor

Luís B. Almeida é Professor Agregado do Departamento de Engenharia Electrotécnica e de Computadores do Instituto Superior Técnico, Universidade Técnica de Lisboa. É também chefe do Grupo de Redes Neuronais e Processamento de Sinais do INESC, Instituto de Engenharia de Sistemas e Computadores.

imagine a estação gráfica mais rápida do mundo. Pode ser sua hoje!

QUANDO O PROBLEMA É A RELAÇÃO PREÇO/DESEMPENHO, E É PRECISO OPTAR POR UMA ESTAÇÃO GRÁFICA UNIX, DESCOBRE-SE QUE A NOVA E EXTRAORDINÁRIA GAMA DE ESTAÇÕES ALPHA AXP ESTÁ UMA GERAÇÃO À FRENTE DA CONCORRÊNCIA.

Agora, a potência e a rapidez dos Alpha AXP aceleram a execução de aplicações tão exigentes como CAD/CAM, engenharia, modelos financeiros, desenvolvimento de software, análise e simulações. Os Alpha AXP não apenas oferecem a maior gama de estações gráficas disponível, mas também lhe proporcionam liberdade de escolha, total capacidade de upgrade, uma integração mais fácil e um desempenho surpreendente (400 MIPS, 161 SPECmark). Toda a potência necessária para manter a liderança do seu negócio!

Contacte-nos pelo telefone 0 500 1 500

digital

imaginação a trabalhar

Digital Equipment Corporation 1993. Digital, o logotipo Digital e Alpha AXP são marcas registadas da Digital Equipment Corporation.

TÉCNICA
revista de Engenharia - 92

comunicação de imagem, codificação de vídeo, normalização, videotelefone, videoconferência, recomendação CCITT H.261, esquemas de codificação híbridos

image communications, video coding, standartisation, videotelephone, videoconference, CCITT H.261 recommendation, hybrid coding schemes

Fernando M. Bernardo Pereira
Secção de Telecomunicações, DEEC, IST, UTL

A STEP FURTHER ON IMAGE COMMUNICATIONS: THE CCITT H.261 STANDARD

Resumo

A comunicação de imagem foi, na última década, uma das mais excitantes e profícuas áreas de investigação, não só devido ao ritmo vertiginoso com que se processaram os avanços teóricos e tecnológicos, mas também devido à dimensão do mercado previsto para os serviços entretanto desenvolvidos. Estes serviços mudarão drasticamente, num futuro próximo, o modo de vida actual.

Entre os paradigmas deste desenvolvimento está o algoritmo de codificação CCITT H.261 para videotelefone e videoconferência a $p\times64$ kbits/s. Este algoritmo foi o resultado de um intenso trabalho de colaboração entre a comunidade internacional ligada à comunicação de imagem.

Este artigo começa com uma breve análise das comunicações de imagem actuais referindo particularmente as técnicas de codificação e os avanços na normalização. Na fase final é dedicada special atenção à norma CCITT H.261; este algoritmo de codificação é descrito m detalhe e o seu desempenho avaliado.

Abstract

Image communication has been, in the ast decade, one of the most exciting and fruitful research fields not only due to the vertiginous theoretical and technological advances but also due to the promising markets foreseen for the services meantime developed. These services will drastically change the nowadays way of life in a very near future.

Among the milestones, there is CCITT H.261 coding algorithm for videotelephone and videoconference at $p\times64$ kbits/s. This algorithm was the result of an intense cooperative work between the whole image communication community.

This paper begins with a brief overview on image communications, referring particularly the image coding techniques and the standardisation advances. Finally, special attention is given to the CCITT H.261 standard; this coding algorithm is described in detail and its performance is evaluated.

1 Introduction

Whatever your best in demagogy, it is difficult to exaggerate the relevance of image for the human being. In the complex field of human relations, communication may be established in many ways but vision plays always a primordial and often decisive role. Popular wisdom states the importance of vision in proverbs as "Out of sight, out of mind"; S. Thomas proclaimed that "Seeing is believing" and Napoleon supported "One image is better than one hundred words". Curiously the image language hides a great paradox since it is at the same time tremendously complex - there are *"messages" impossible to transmit with other languages* - and encouragingly simple - *try other ways of communication between a papuan and an eskimo.* Whatever the opinion on its complexity there are hardly any doubts about its power. Who can forget the impact of the images of the Vietnam war in the USA people or the images of the Ethiopia' hungry children for the development of the 'Live Aid' world campaign? Would the world be the same without images?

Defining image communication as the transference of image information across time, space or both, we easily conclude that communications engineering is a very old job. The concept ranges from the conventional television broadcasting or audiovisual communications - *where space is conquered* - to the storage of image information - *where time or time and space are mastered.* Looking at the evolution of Telecommunications as a continuous emulation of the complex human communication systems, it is possible to conclude that image communications are not just one more step but rather a fundamental one to reach this aim [1].

Having in mind that the first really generalised image communication system - *Television* - changed, for the best and for the worst, our civilisation, by entertaining, disseminating ideas and stimulating comparisons, the time appears to be ripe to move one step further - *point to point audiovisual communications.* These new communication systems range from videotelephony and videoconference to photographic videotex or cable TV by request and will spur a large number of applications with significant social impact. One simple but significative example are the new frontiers opened by videotelephony for the communications, care and independence of old or handicapped people. Point-multipoint communications also present new developments beginning with the already famous High Definition Television (HDTV). The success of the new services will depend not only on the quality-price ratio but also on their flexibility and interoperability. In fact most of these services require a suitable infrastructure which it is not immediately available although will be in the near future, at least for the more developed regions.

One of the technical reasons that until now barred the development of audiovisual communications has been the large bandwidth required. In the early eighties, the development of digital techniques, the research efforts on image coding, the intention of

Original submetido para publicação em 20 de Março de 1992

many administrations to provide video services through the Integrated Services Digital Network (ISDN) and the first proposals and plans for the future Broadband-ISDN brought out the first audiovisual communication experiments. Lately, with the explosion of digital technologies, image coding assumed an increasing relevance since efficient coding methods allow the reduction of the transmission rate and the transmission costs for a given picture quality. The ultimate goal in the design of an image compression system is to minimise the bandwidth requirements for the transmission of a specified quality with relatively low-cost, compact and high-speed very large scale integration circuits. During the last decade a set of concepts and coding algorithms were standardised - *e.g. digital studio format CCIR 601, videotelephone and videoconference coding algorithm CCITT H.261.* A few others are likely to follow suit in the near future - *photographic videotex coding algorithm ISO/JPEG, digital storage media coding algorithm ISO/MPEG.* With the explosion of new techniques and methods in all scientific areas related to communications, standardisation organisations assume a fundamental role of coordination and even steering thus avoiding the birth of a chaotic communications market with all its negative effects. The new standards and equipments are already changing the outline of video communications and announcing the advent of multiple new applications with hard to foresee economical, social and even ethical and philosophical impacts.

While a generation of image communications technology is beginning to prove its excellence, another generation is under way in many of the world research centres - *the Broadband-ISDN or B-ISDN, as it is often known.* B-ISDN represents the most recent evolution of Telecommunications and aims to provide, in the metropolitan and long distance areas, high speed multimedia communications and information services. This service improvements make use of optical fiber technology, high speed and high capacity switching equipments and signal processing techniques to answer the emerging demand for broadband services. B-ISDN is directed towards an integrated services transport mechanism - *interactive and distribution services* - integrating both circuit and packet transfer modes into one universal broadband network. B-ISDN is presented as an ISDN development achieved by the progressive incorporation of additional functions and services such as high quality video [2]. One of the most innovating B-ISDN features is the Asynchronous Transfer Mode (ATM), a packet oriented transfer mode using asynchronous time division multiplexing where the multiplexed information flow is organised in fixed size blocks called cells. The transfer capacity is assigned on demand at the call setup depending on the source characteristics and on the available resources.

The flexibility of ATM environments opens new opportunities for video communications, up to now limited by the channel characteristics. The challenge is the efficient use of the network resources to achieve acceptable quality rather than to find the best tradeoff between the available resources and the final quality. This new environment needs video coding algorithms which exploit the network capabilities [3] - *e.g. variable bit rate coding* - and that overcome the new types of impairments - *e.g. cell losses.*

2 About image coding

According to Habibi [4], image coding is concerned with the conversion of an analog picture into the smallest set of binary integers such that it can be used to reconstruct a replica of the original signal. Of course this must be achieved simultaneously with the overall goals of the system (often only vaguely defined) or may also involve other considerations such as viewer satisfaction. In conclusion, and as support Netravali and

Limb, "(...) the application of picture coding to transmission channels is an economic tradeoff in system design, balancing picture quality, circuit complexity, bit rate and error performance." [5].

Although early efforts on image coding used analog techniques to reduce bandwidth - *bandwidth compression* -, almost all recent coding techniques are directed towards digital transmission. This is due to digital many advantages, namely the flexibility, the possibility of regeneration, the easy multiplexing and encryption and finally its increasing diffusion [5]. Image coding exploded in the early eighties when the capacity, speed, compactness and price of digital technology became competitive. For the first videoconferencing experiments, satellite transmissions, image storage and military applications, image coding provided the means to reduce the costs noticeably. The progressive development of digital communication networks for speech, text and data generated a growing demand for the addition of image transmissions. This general context led to an increasing interest of the PTT's on image coding to provide video services using the ISDN basic access limited channel capacity and spurred the interest for a Broadband-ISDN where service integration, including all types of video services, is the main feature. The continuous and rapid evolution of digital technologies in the last decade fostered an incredible amount of research and development work on image coding which already led to some international standards. Standardisation activity, which will be briefly analysed in the following, assumes now a decisive role in the coordination and efficient use of the scientific work. It is trying to prevent the image coding 'adolescent' phase to create higher barriers than analog technologies.

2.1 Some remarks on image coding techniques

Back in 1982 when the CCIR recommended the digital video standard for the handling of studio colour TV with a data rate of 216 Mbit/s [6], the relevance of image coding for the future of digital TV became clear. Image coding techniques must take advantage of the considerable amount of superfluous information produced by the traditional coding of visual information. The superfluous information may be divided in: statistical redundancy and subjective redundancy or irrelevancy [7].

Statistical redundancy is related with the similarities, correlation and predictability of data. Since statistical redundancy reduction does not involve any loss of information, the quality of the image is not degraded and it is possible to recover the original image. Subjective redundancy relates to the information unperceived by the human eye or which the human brain will find insignificant. Unlike statistical redundancy the loss of irrelevancy is irreversible and the original image can no longer be recovered. Since image coding deals with redundancy and irrelevancy, it is essential to understand the visual information statistics and the characteristics of human vision.

The image coding problem may be mathematically formulated in the context of information theory as the research of the encoding strategy which minimises an average distortion between the original and the coded signals [5]. The complexity of this analysis, due to the lack of statistical models for picture signals and to the absence of a recognised distortion criteria taking into account the human visual system, urges a pragmatic approach. During the last decades many coding methods have been suggested which exploit different kinds of redundancy in picture signals. A clear classification of these techniques is somewhat difficult due to the great variety of principles referred to [4,7,8,9,10]. Moreover almost all the methods use an adaptive technique where the coding parameters change as a function of the data. Factors to account for human perception and to improve the overall picture quality are used in many methods (see the

luminance/chrominance spatial resolutions or the threshold matrices of the ISO/JPEG and ISO/MPEG algorithms [11, 12]). Over the past years a number of mathematical models of the human visual system (HVS) have been suggested for image processing applications. However the introduction of such factors on image coding must take into account the different services coding/quality requirements as well as the 'higher efficiency/higher error sensitivity' statistical phenomenon.

Netravali and Haskell [7] classified the main coding techniques making use of five categories of waveform coding and one category for statistical coding, all of which are briefly referred to below. One should however recall that a coder can use more than one coding technique in order to produce the 'winning cocktail' for the required cost/performance constraints (see the CCITT H.261 standard).

2.1.1 Pulse code modulation (PCM)

Pulse Code Modulation is a time discrete - *usually at the Nyquist rate* - and amplitude discrete - *usually 256 levels* - representation of visual information. PCM is no longer looked as a coding technique due to its low statistical and subjective redundancy reduction capacities. However it has an important function as intermediate representation step prior to the application of more complex coding schemes. PCM representation has also a role as the reference quality for subjective quality analysis.

2.1.2 Predictive coding

Predictive coding, also known as Differential Pulse Code Modulation (DPCM), predicts the sample to be coded using the values of the previously coded samples and only codes the differential error. Coding efficiency increases therefore with the accuracy of the predictions. Typical cases of predictive coding are:

- *Traditional interframe coding where pixels of frame k are coded using as prediction the coded values of the corresponding pixels in frame k-1.*
- *Conditional replenishment where the differences, either spatial or temporal, between the predicted and actual values (also called errors) are transmitted only if above a chosen threshold.*

The best known drawback of predictive coding is error propagation which requires some kind of periodic non temporally correlated coding.

One important way of improving the prediction accuracy is the use of local and/or global motion compensation as in the CCITT H.261 standard or in the ISO/MPEG algorithm. DPCM is also used in the ISO/JPEG coding algorithm to provide a perfectly reversible coding mode.

2.1.3. Transform coding

Transform coding converts a sequence of statistically-dependent picture elements into an array of relatively independent and information-compacted transform coefficients [7, 13, 14]. Data compression of image information is achieved in the transform domain by non linear quantization or by uniform quantization followed by entropy coding. Transform coding uses unitary or orthonormal transforms applied to the entire image or repeatedly to many identical subsections of the image, known as blocks. Block dimension is a tradeoff between computational effort and data compression due to image spatial correlation. All unitary transforms share two important properties: the invariance of entropy and power.

Transform coding is adequate for low bit rates achieving good image quality. All major actual coding standards use transform coding particularly Discrete Cosine Transform (DCT) which is the best approach to the ideal Karhunen-Loeve transform (KLT) for tightly correlated signals such as image signals.

A well known solution for image coding is hybrid coding - *CCITT H.261, ISO/MPEG, ISO/JPEG* - which uses transform and predictive codings, in principle in any order, but usually through the transform of temporal sample differences.

2.1.4. Interpolative/extrapolative coding

Interpolative and extrapolative coding is based on a subset of image pixels from which the remaining pixels are obtained by interpolation or extrapolation. Although this technique has seldom been used, due to some services and hardware implications, it recently found an important application in the ISO/MPEG coding algorithm for moving video on digital storage media [12]. This future ISO standard achieves a substantial quality improvement, compared to the similar CCITT H.261 algorithm, essentially due to the introduction of temporal interpolation integrated with motion compensation. The drawback of this technique is the absolute necessity of transmitting the non-interpolated frames out of order which increases the hardware complexity and introduces a higher initial delay.

2.1.5. Entropy coding

The basic principle of entropy coding is the assignment of the optimum code word (and code length) for each coder output symbol depending on its statistical distribution. A well known example is Huffman coding, also known as VLC (Variable Length Code) coding, where the boundaries between adjacent code words can be automatically deduced from the transmitted sequence of bits - *comma-free* - in the absence of transmission errors. A more recent example of entropy coding is arithmetic coding which is optional to the baseline system of the future ISO/JPEG photographic videotex coding standard [11].

While previously presented coding methods are irreversible, essentially due to the quantization operations, entropy coding preserves information (reversible) and is nowadays a fundamental element in all practical image coding schemes.

2.1.6. Other coding techniques

The list of image coding techniques not included in the previous classification is long. Besides coding techniques specially adapted to particular types of images, as contour coding for bilevel signals, this category includes among other vector quantization and pyramidal and subband coding.

Run length coding is a very general technique which assigns a code word to each continuous group of identical pixels or bits or any other event - *run* - taking into account the statistical distribution of the runs. This method is used in facsimile standards (CCITT T.4 and T.6) but also in the CCITT H.261 standard to code the runs of zeros between two non zero DCT coefficients.

One characteristic of almost all the coding techniques is that quantization is performed on individual real-valued samples; such is the case for the transform coefficients and for the differential error of predictive techniques. These techniques are non optimal since the processed samples are already somehow correlated or dependent. According to Shannon's rate distortion theory, a better performance is always achievable in theory by coding vectors instead of scalars, even though the data source is memoryless [15].

Vector quantization performs range partitioning and code value selection using 2^{NR} elements of a N-dimensional vector space. The vector space is partitioned into 2^{NR} subsets each with a representation value or code vector. Each N pixels block is coded with one code vector - NR bit/block or R bit/pixel. The goal of vector quantization is to obtain a quantizer consisting of 2^{NR} range-word pairs minimising the expected distortion for the class of images to be handled. In other words the fundamental questions in vector quantization are the definition of the distortion criteria optimum partitioning ranges and corresponding code vectors [16].

Pyramid and subband coding are based on the representation of the uncoded image as a series of band-pass images [17, 18]. While pyramid coding uses a series of low-pass filtered images sampled at successively lower rates and codes the error differences after interpolation, subband coding uses the output filtered images of a bank of n parallel band-filters. The specific coding method of each filtered image depends on the statistical characteristics of the image signal, on the relevance for the final quality of each band and on the available data resources.

In the last years, 'subband coding' gained a general meaning referring not only to the coding with a bank of band-pass filtered images but to any kind of coding where the signal is divided in 'bands' independently of the 'band' criteria definition; an example may be the division in 'bands' of a DCT coefficients block.

2.1.7. New trends on image coding

Basic research in the image coding field progressed along several new routes, in the last years. The most promising ones appear to be related to the definition of new image description models such as Model Based coding and Fractal Based coding [19].

In general one can distinguish between two model based coding cases, depending whether the scene contains known or unknown elements. A typical application of the first case is videotelephone 'head-and-shoulder' images where the goal is to reach very low data rates, eventually lower than 24 kbit/s. The second alternative considers the description of motion and shape of 3-dimensional objects in order to improve the performance of existing low data rate coders. Some of the description techniques are 3D solid modelling, modelling of facial actions, 3D motion estimation, locating facial features and real time image synthesis. The complexity of most of the available algorithms to perform these functions is not acceptable for real time implementations but results are promising.

The synthesis of complex scenes using fractals has became popular in computer graphics and recently invaded the area of natural images. Since fractals may be described by a small number of parameters, they may lead to substantial data compression if the initial 'artificial aspect' is eliminated. Though it is quite simple to generate a fractal image, the inverse problem, e. g. the generation of the fractals for a chosen image, is much more complicated. Developments in these areas are sometimes surprising as the recent presentation in the USA of a fractal codec for still and moving images [20].

A closer relation between image coding and image synthesis allowing higher compression factors is foreseen in the near future. As Prof. Forchheimer from Linköping University - Sweden said in the recent Workshop on 64 kbit/s Coding of Moving Images (1990), "Here is definitely the area where many coding people will make their contributions during the next decade." [19].

3 Image coding standardisation activities

The relevance of the standardisation process for a fruitful and efficient development of video communications is beyond discussion. A typical example of the consequences of the lack of standards is facsimile which remained unknown for many years until the great explosion after the standardisation of the various facsimile groups.

The fight for image communication standards started many decades ago in CCIR with television systems. Although at the beginning the competencies of each international standardisation body were clear - *TV to CCIR since uses radio transmission and videotelephony to CCITT as an extension of telephony and telegraphy* - the development of image communications and the increasing demand from customers/subscribers blurred the previous frontiers and pressed the demands for a growing coordination process. This fact is substantiated by the setting up of joint committees and appropriate links between the various standardisation organisms trying to prevent later incompatibilities and to avoid the proliferation of studies with duplication of efforts.

Standardisation on image coding started in the early eighties when digital technology became a serious option. The international organisms and study groups involved in standardisation process are many - *CCIR, CCITT, ISO, IEC, ETSI, CIE, CMTT, EBU, COST, RACE, EUREKA, etc.* In the field of image communications, three main areas with some links must be considered [21,22]:

- TV and HDTV
- Real Time Visual Communications
- Image Based Telematic Services

3.1. TV and high definition TV

The first international recommendation on digital TV was Recommendation CCIR 601 with the "Encoding Parameters of Digital Television for Studios" [6]. This recommendation defined the signal components - *one luminance and two colour difference signals* - as well as the sampling frequencies, quantization levels and active picture areas. The definition of a similar format for HDTV has been impossible until now essentially due to the existence of two different proposals from Japan and Europe. One can foresee another dual or even triple solution (depending on the USA position) thus loosing once again the opportunity for a worldwide TV distribution format [23].

When discussing TV and HDTV codings, one must consider three different aspects: broadcasting, recording and point-to-point transmission.

Though NTSC/PAL/SECAM signals may be used for satellite broadcasting, the availability of a wider baseband channel - *8 to 11 MHz* - brought about the development of a new TV standard - *MAC (Multiplexed Analog Components)* - specially adapted to direct satellite broadcasting. In the interim Japan launched an ambitious project to bypass the drawbacks of existing TV systems in order to satisfy human eye and brain requirements - *HDTV*. The Japanese system - *MUSE (Multiple Sub-Sampling Encoding)* - ignores the existing status and uses a complex analog coding - *motion adaptive, sub-sampling and motion compensation* - in order to allow for the limited RF satellite band for direct broadcasting - *24 to 27 MHz*. European reaction led to the proposal of a production and broadcasting standard with analog coding - *HD-MAC (motion adaptive, sub-sampling and motion compensation)* - compatible with the MAC system. Both are under discussion in the CCIR Study Group 11- Broadcasting Service.

TV and HDTV recording are still strongly confined to analog technology. However, while the CCIR is preparing recommendation 657 with the definition of the video re-

corder - *D1 VTR* - corresponding to the CCIR 601 digital studio format, HDTV is significantly affected by the lack of a standard format. Nevertheless several manufacturers produce recording and display equipment according to MUSE specifications.

The technical and economic feasibility of carrying hundreds of Mbit/s to the home using optical fibbers and the definition of 155 Mbit/s as the basic channel for the B-ISDN with a net data rate of 145 Mbit/s, stimulated the study of coding algorithms able to transmit good quality TV and HDTV signals within these data rate limitations. CCIR and CMTT are now working on two coding algorithms for 34 and 45 Mbit/s using predictive and transform coding techniques. For HDTV the situation is less clear (there is also a two-layer coding alternative) but a final goal between 34 and 140 Mbit/s should be reached. An experimental HDTV satellite/optical fiber link has been established by RAI for the Football World Cup' 90 using a hybrid DCT codec working from about 60 Mbit/s to 140 Mbit/s. The studies going on within ISO/MPEG 2 project will have particular relevance in the future of digital TV/HDTV.

3.2 Real time visual communications

Although there were some early attempts to introduce image on basic telephony - *e.g. Bell system in 1968* -, only digital technologies had enough impact to give the final push to videoconference and videotelephony [24]. This new class of services represents the incoming generation of video communications where general point to point visual transmission became available as another step towards the 'global village'.

The first digital image coding standard - *CCITT H.100 series recommendations in 1984* - uses the available primary rate digital lines - *1544 and 2048 kbit/s* - and is basically a conditional replenishment scheme where the 'moving parts' are coded with variable length DPCM [25]. The overflow of the buffer in the coder is recovered discarding first one sample out of two and, in extreme cases, one field out of two; underflow is recovered with systematic updating. The H.120 standard uses a temporal resolution of 25 frame/s with 286 lines interlaced; each line carries 256 luminance samples and 51 samples of one chrominance, C_R or C_B, on a line alternating basis. Recommendation H.140 also defines the H.120 compatible procedures to implement multiconference.

Due to the continuous and fast evolution of low bit rate video coding during later years, when H.100 series recommendations were adopted, new powerful coding methods were available. In addition the foreseen ISDN basic access data rates - *128 kbit/s of data and 16 kbit/s of signalling* - required a more powerful coding standard in order to achieve ISDN video transmissions.

This context led to the creation by the CCITT of the 'Specialists Group on Coding for Visual Telephony' to standardise the second generation sub-primary rate video codecs - *nx384 kbit/s and px64 kbit/s* [26]. This group produced the CCITT H.261 standard [27] milestone on image coding due to its efficiency and the creation of a compatibility issue that new algorithms must consider. The H.261 standard is a motion compensated DCT hybrid scheme using the CIF resolution format (360×288 pixels for luminance and 180×144 pixels for each chrominance).

While the H.261 standard is spreading worldwide waiting for a consistent ISDN diffusion, a new step in real time video communications is being prepared with the future B-ISDN [2,28]. Since this network will provide more flexibility allowing a closer adaptation between the coding algorithm and the signal, a new category of image coding algorithms must be created.

3.3 Image based telematic services

The need to improve videotex performance was one of the reasons for the creation by ISO and CCITT of the Joint Photographic Experts Group (JPEG) directed to the standardisation of a coding algorithm able to provide good quality photographic (still) pictures. For videotex - *access to remote data bases via the telecommunications network* - an efficient way to code single multilevel or colour pictures is essential. Many other interesting applications may be found for this algorithm which is already in the final phase of the standardisation process. The future JPEG standard uses ADCT (Adaptive DCT) applied on 8x8 blocks and provides three operating coding systems: the baseline, the extended and the lossless [11].

The introduction of moving images on videotex or simply the provision of a video coding algorithm for digital storage media were behind the creation, also by ISO, of the Moving Pictures Experts Group (MPEG) to standardise a coding algorithm for digital video recording with a bit rate up to about 1.5 Mbit/s [12].

4 CCITT H.261 video coding standard

After the H.100 series recommendations (1984) which considered two options, *1544 kbit/s and 2048 kbit/s*, the work continued on a coding standard for sub-primary data rate codecs. In the same year, the CCITT Study Group XV established the 'Specialists Group on Coding for Visual Telephony' to standardise the second generation sub-primary rate video codecs - *nx384 kbit/s and px64 kbit/s*. Recommendation CCITT H.261 has been included for the first time in the CCITT Blue Book in 1988 (Melbourne) leaving some parameters open to future definition. This work was finished for px64 kbit/s in 1990 [27]. In all the standardisation process, the work of the european joint project COST 211, with its fundamental contributions to the achievement of the final goals, was very important.

As the first coding standard of a second generation of codecs using the most efficient coding techniques, the H.261 standard has a relevant position in the video communications outline and creates a new compatibility issue. This new constraint was present in ISO's work for the standardisation of a codec for moving video on digital storage media and it is fundamental for the recent research of efficient coding algorithms for the future B-ISDN. As could be expected, not all the coding tools are completely specified; only those essential for a perfect coder/decoder synchronism are included. Interesting examples are the decision and the representation levels of the quantizer characteristic where the former are 'free' and the later standardised. However, even if some of the tools are not completely specified, they are in a certain sense limited since an efficient coding requires a real statistical distribution similar to that used for the VLC tables, such as the motion compensation decision characteristic which is not standardised but it is strictly related to the motion vectors VLC's.

In the following some of the most relevant H.261 characteristics will be described. For all the tools not completely specified, the options of the COST 211 bis "Reference Model" [29], recognised simulation environment of the H.261 scheme, will be used. This description is not complete and does not attempt to replace the text of the recommendation.

4.1 Motivations

According to the H.261 draft revision [27], the main CCITT reasons for this recommendation were:

- *The significant demand for audiovisual services, namely videotelephone and videoconference, and the availability of digital circuits able to satisfy this demand, namely at the B, H0 or H11/H12 rates - 64, 384 and 1536/1920 kbit/s.*
- *The near future availability in some countries of ISDN links providing a switched transmission service at the B, H0 and/or H11/H12 rates and the need of providing some means of intercommunication between the new likely audiovisual services.*
- *The statement that recommendation H.120 for videoconferencing using primary digital group transmission was the first in an evolving series of recommendations.*
- *The advances in research and development of video coding technologies, namely for low bit rates.*

Recommendation H.261 describes the video coding and decoding methods for the moving picture components of audiovisual services at rates of px64 kbit/s, where p ranges from 1 to 30.

4.2 Spatial / Temporal Hierarchical Structure

The image sequence is organised in a hierarchical structure with four layers:

1) Picture

The source coder operates on pictures using the CIF or QCIF formats (half vertical and horizontal CIF resolution). These formats result from a compromise between the processing efforts of the various TV systems to reach an intermediate common format. All codecs must be able to work with the QCIF resolution and some codecs can also operate with CIF.

The coder operates on non-interlaced pictures with a temporal resolution of 30 Hz. Other temporal resolutions are allowed with at least 1, 2 or 3 non-transmitted pictures between each transmitted one. Pictures are coded with one luminance and two chrominance signals, as defined by Recommendation CCIR 601.

2) Group of Blocks (GOB)

Each picture is divided into Group of Blocks of similar dimension. A GOB comprises one twelfth of the CIF frame - *distributed as indicated in figure 1* - or one third of the QCIF frame. One GOB corresponds to 176 active pixels by 48 lines for luminance and the spatially corresponding 88 pixels by 24 lines for each of the chrominances.

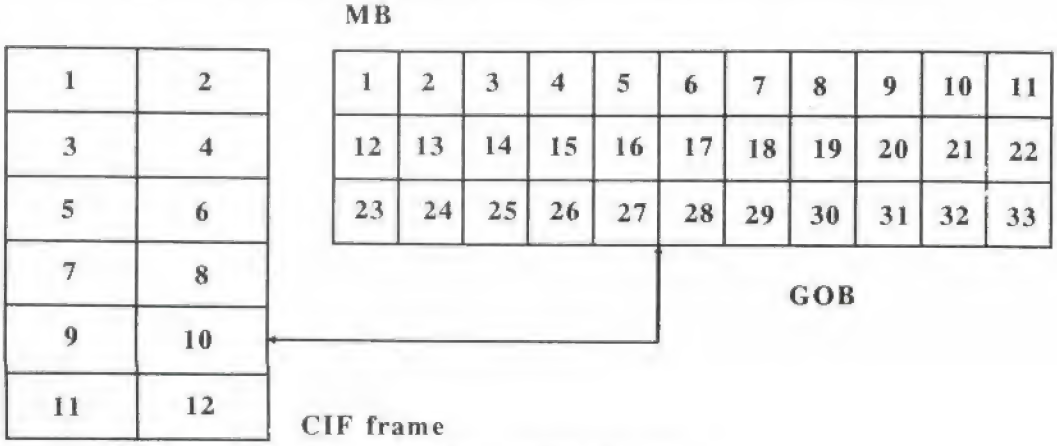


figure 1 Group of Blocks (GOB)

3) Macroblock (MB)

Each GOB is divided into 33 macroblocks, as indicated in figure 1. A macroblock comprises 16 pixels by 16 lines of luminance and the spatially corresponding 8 pixels by 8 lines of each of the chrominances (figure 2). The position of a transmitted macroblock inside the GOB is given by the Macroblock Address - *MBA*. While for the first transmitted MB in a GOB, MBA is the MB absolute address, for the subsequent MB's, MBA is the difference between the absolute address of the actual MB and that of the last transmitted MB - *relative addressing*.

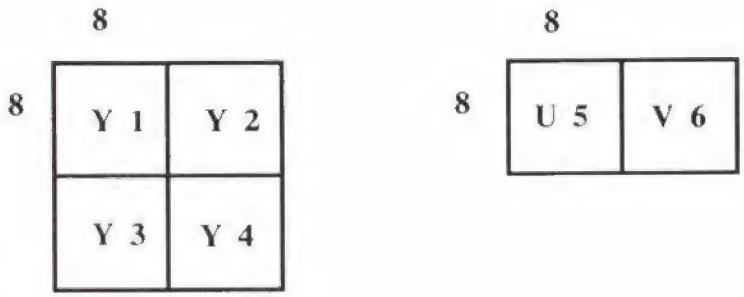


figure 2 Macroblock (MB)

4) Block

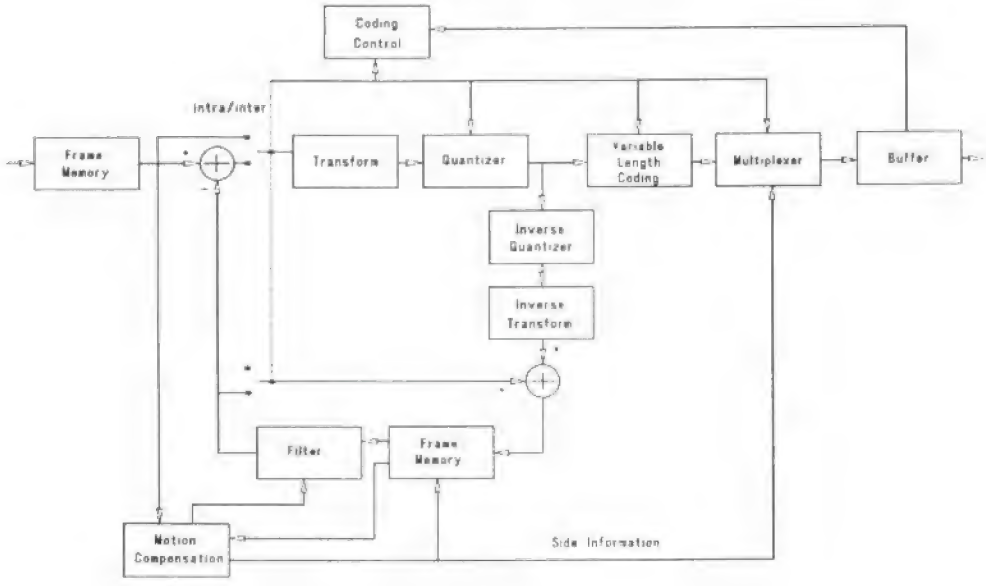
The block is the lower layer of the spatial coding hierarchy and comprises 8×8 pixels, that is 8 pixels per line on 8 consecutive lines. A MB corresponds to four blocks of luminance and two blocks of chrominance, one per component, as shown in figure 2.

4.3 Video source coding algorithm

The general architecture of the H.261 video coding algorithm is given in figure 3. The main tools of this coding scheme are prediction, block transform and quantization. The H.261 standard is a block based scheme (8×8) where the prediction error - *inter modes* - or the input picture - *intra mode* - is transmitted after transformation, quantization and entropy coding. The algorithm tries to exploit the image spatial correlation by using the macroblock concept.

4.3.1 Prediction

Predictive coding is basically directed to make use of the image temporal correlation. In the H.261 algorithm each frame is interframe predicted that means each sample is predicted using the corresponding previous sample. Since coding efficiency increases with prediction accuracy, it is important to improve the prediction introducing the motion compensation and a spatial filter. For macroblocks with very low temporal correlation inter coding may be substituted by an intra mode where information is coded using only the macroblock internal correlation.



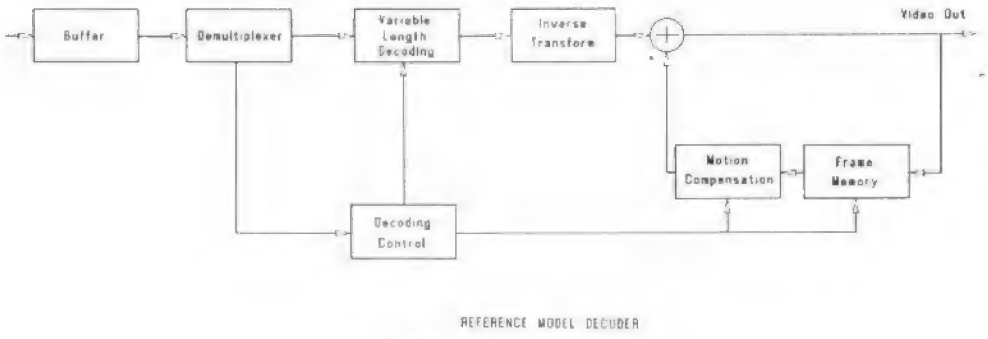


figure 3 H.261 codec architecture

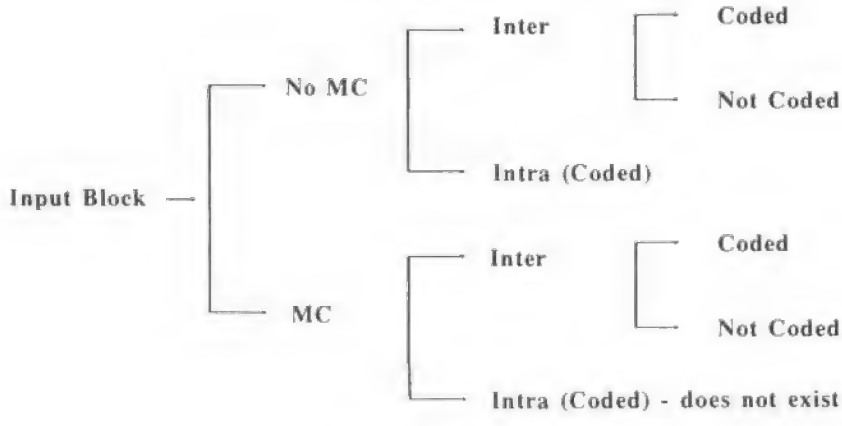


figure 4 Reference Model mode decision tree

The coding mode is macroblock decided and the decision criteria is not included in the recommendation. Figure 4 shows the Reference Model decision tree where the motion compensation / spatial filter are coupled operations; the 'coded' and 'not coded' attributes indicate if transform coefficients are transmitted or not. The macroblock type - *MTYPE* - may also indicate the successive transmission of a word - *Coded Block Pattern (CBP)* - containing a pattern number related to the blocks in the macroblock for which at least one transform coefficient is transmitted. This pattern is computed as

$$CBP = 32\,P_1 + 16\,P_2 + 8\,P_3 + 4\,P_4 + 2\,P_5 + P_6$$

where P_n is 1 if any coefficient is transmitted for block n, and 0 otherwise.

4.3.1.1 Motion compensation

Motion compensation (MC) is an optional coding tool for the coder. The motion compensation decision criteria and the choice of motion vectors are not recommended. However some basic rules must be followed:

- *One motion vector per macroblock may be transmitted.*
- *The range of horizontal and vertical motion vector components is +15 to -15, integer values only.*
- *The transmitted macroblock motion vector is used for the four luminance blocks. The chrominance motion vector is obtained by halving the luminance vector and truncating the magnitude towards zero to yield integer components.*
- *A positive value of the horizontal or vertical motion vector components means that the prediction must be done using the pixels in the previous frame which are spatially to the right or below the pixels being predicted.*
- *Only motion vectors referencing existing areas of the frame are valid.*
- *A differential motion vector, computed as the difference between the actual motion vector and its prediction which is the motion vector of the previous macroblock, is transmitted. The motion vector prediction is taken as zero if:*
  - *the actual macroblock is 1, 12 or 23;*
  - *the last transmitted macroblock is not contiguous to the current one;*
  - *the previous macroblock was not motion compensated.*

The COST 211 Reference Model (RM) motion compensation decision characteristic, shown in figure 5, may be adopted. In this figure **bd** is the sum of the block differences and **dbd** is the sum of the displaced block differences. MC off includes the separation line. The number and order of the candidates to MC prediction macroblock is not defined and may depend on the available processing power. The Reference Model uses the 3-step method - *range +7 to -7* - but in the digital storage media video coding algorithm a full search is used. This problem is becoming less critical with the recent development of full search motion detection chips.

4.3.1.2 Loop filter

The prediction may be improved by the introduction of a two-dimensional spatial filter which operates on a picture block. This filter may be useful to reduce the high frequency components due to MC and/or to quantization noise in the feedback loop. The loop filter is separable into two non-recursive one dimensional horizontal and vertical

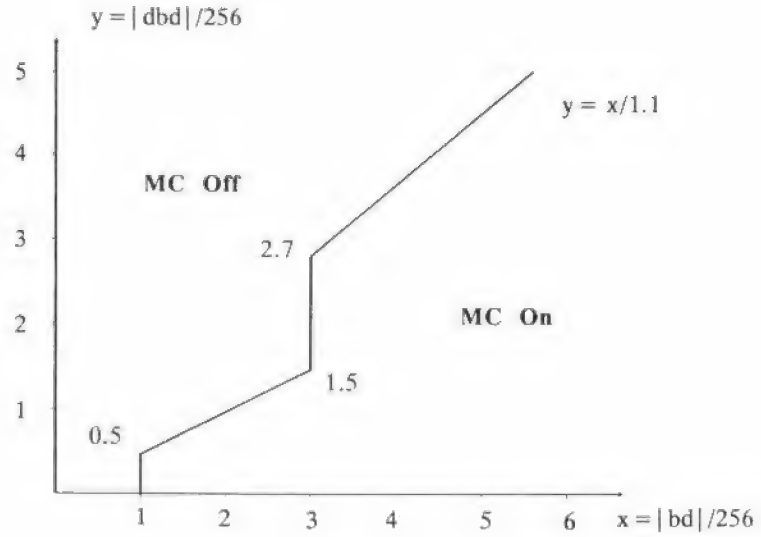


figure 5 RM motion compensation decision characteristic

filters both with coefficients of 1/4, 1/2, 1/4, except at block edges where the coefficients are changed to 0, 1, 0. The loop filter is either applied to all 6 blocks of a macroblock according to the macroblock type or not at all.

4.3.1.3 Intra mode

The intra coding mode has been introduced to improve the algorithm performance in situations such as scene cuts, fast movements or areas of decovered background. To control the accumulation of inverse transform mismatch and break error propagation, some kind of periodic intra coding is forced. The pattern of this forced updating is not defined but a macroblock must be updated at least once every 132 times it is transmitted. Since the intra/inter decision characteristic is not subject to recommendation, the Reference Model characteristic may be applied [29].

4.3.2 Transform

The prediction error or the intra block information is processed before transmission by a separable two-dimensional Discrete Cosine Transform (DCT), applied to 8×8 blocks. The inverse transform output is clipped from -256 to +255 to allow representation with 9 bits. The DCT transfer functions are shown in figure 6. Though the arithmetic procedures to compute the DCT coefficients are not subject to recommendation, the inverse transform must meet the error tolerance specified in Recommendation H.261 - Annex 1, in order to avoid mismatch problems.

While for intra coded macroblocks all the 6 blocks have transmitted DCT coefficients, for inter coded macroblocks only the blocks indicated by the MTYPE and the CBP have transmitted transform coefficients. Since the DCT transform compacts the signal energy on the upper-left coefficients region, the quantized DCT coefficients are sequentially transmitted following a zig-zag scanning. The coefficients are transmitted as (RUN, LEVEL) pairs, where RUN is the number of zero coefficients after the previous transmitted coefficient and LEVEL is the quantization level of the actual coefficient. Coefficients after the last non-zero one are not transmitted; all blocks with transmitted coefficients end with the special word End Of Block - *EOB*.

$$F(u,v) = \frac{1}{4} C(u)\, C(v) \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos\left[\frac{\pi u(2x+1)}{16}\right] \cos\left[\frac{\pi v(2y+1)}{16}\right]$$

where: $u = 0,1,2,\ldots 7$
$v = 0,1,2,\ldots 7$ Transform coordinates

$$f(x,y) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C(u)\, C(v)\, F(u,v) \cos\left[\frac{\pi u(2x+1)}{16}\right] \cos\left[\frac{\pi v(2y+1)}{16}\right]$$

where: $x = 0,1,2,\ldots 7$
$y = 0,1,2,\ldots 7$ Spatial coordinates

$$C(u),C(v) = \begin{cases} \frac{1}{\sqrt{2}} & u,v = 0 \\ 1 & \text{else} \end{cases}$$

figure 6 Two-dimensional (8x8) DCT transfer functions

4.3.3 Quantization

The quantization characteristic is linear and defined by the quantization step chosen between the even values in the range 2 to 62. While for the AC coefficients a linear characteristic with a central dead-zone around zero is used, the intra DC coefficients use a linear characteristic with no dead-zone and a step size of 8. Within a macroblock the same quantization step is used for all the coefficients but the intra DC. Due to the limited range of allowed coefficients levels, the full dynamic range of the transform coefficients cannot be represented, specially for the lower quantization steps. The quantization step is transmitted every GOB but may be overridden by a new value transmitted at the macroblock level.

The H.261 quantization characteristic has clearly defined reconstruction levels (REC) but the decision levels are not recommended. Figure 7 presents the Reference Model quantization characteristic where usually $T=g$ (e.g. for $g \leq coef < 2g$, $REC=1.5g$). Under

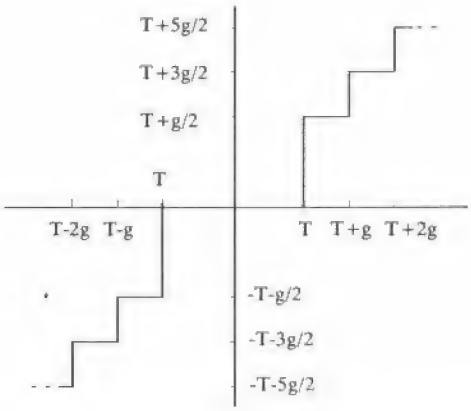


figure 7 Reference Model quantization characteristic

Since the DCT coefficients have different coding relevance, it may be interesting to implement a variable threshold strategy to increase the number of zero coefficients, independently of the quantization characteristic. The Reference Model implements a variable threshold strategy based on the length of zero coefficient strings, one-dimensionally looked after a zig-zag scanning.

4.3.4 Statistical coding

The exploitation of the statistical or entropy coding concept is a must in the H.261 standard. Following this concept, VLC tables are provided to code almost all the transmitted information, namely:

- *Macroblock Address - MBA*
- *Macroblock Type - MTYPE*
- *Differential Motion Vectors - MVD*
- *Coded Block Pattern - CBP*
- *DCT coefficients using the (RUN, LEVEL) concept*

4.4 Coding control

No particular coding control strategy is advised in Recommendation H.261 even if various techniques/parameters are presented as control candidates - *pre-processing, quantization, macroblock type criteria, temporal subsampling of complete frames.* The output bitstream must comply with the requirements of the Hypothetical Reference Decoder defined in Recommendation H.261 - Annex 2.

The Reference Model uses the quantization step to control the rate of generation of coded video data (px64 kbit/s). This technique, generally recognised as being particularly simple and efficient, creates a direct relation between the coder buffer fullness, dependent on the data production, and the quantization step computation by:

$$QS = 2\ \mathrm{Int}\,[Buffer / (200\ p)] + 2$$

where *Buffer* denotes the buffer fullness. The minimum and maximum values of *QS* are 2 and 62, respectively; the RM buffer size is px6.4 kbit. When the buffer is full, the macroblocks are classified as 'Not Coded, No MC' - zero coefficients, no motion vectors (see figure 4). The buffer fullness is updated every macroblock.

4.5 Video multiplex coder

The video multiplex arrangement follows the spatial/temporal hierarchical structure already presented. Syntax diagrams of the various layers are given in figures 8 to 11 - *squared blocks for the fixed length words and round edge blocks for the variable length ones.*

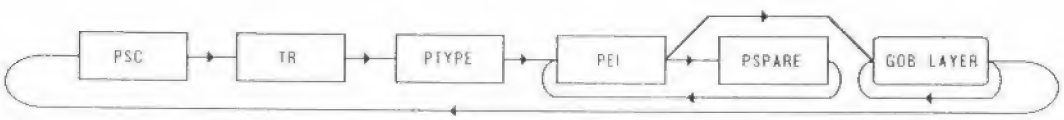


figure 8 Video Multiplex - Picture Layer

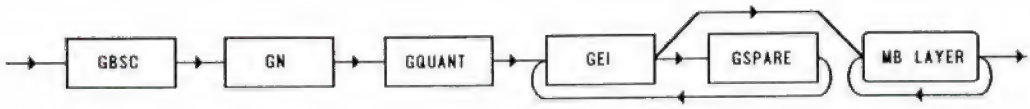


figure 9 Video Multiplex - GOB Layer

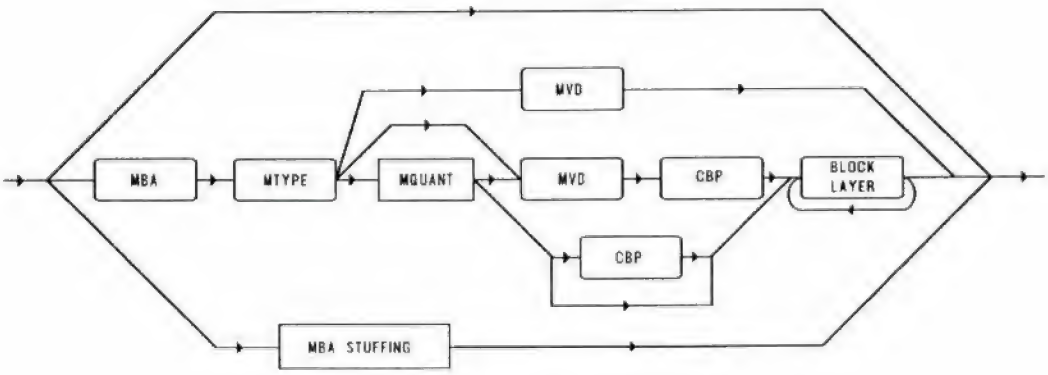


figure 10 Video Multiplex - Macroblock Layer

figure 11 Video Multiplex - Block Layer

The elements of the H.261 video multiplex coder are:

Picture Layer

Picture Start Code - PSC - 20 bits
Temporal Reference - TR - 5 bits
Picture Type Information - PTYPE - 6 bits
Picture Extra Insertion Information - PEI - 1 bit
Picture Spare Information - PSPARE - 0/8/16 ... bits

Group of Blocks Layer

Group of Blocks Start Code - GBSC - 16 bits
Group Number - GN - 4 bits
GOB Quantizer Information - GQUANT - 5 bits
GOB Extra Insertion Information - GEI - 1 bit
GOB Spare Information - GSPARE - 0/8/16 ... bits

Macroblock Layer

Macroblock Address - MBA - variable length
Macroblock Type Information - MTYPE - variable length
Macroblock Quantizer Information - MQUANT - 5 bits
Motion Vector Data - MVD - variable length
Coded Block Pattern - CBP - variable length

Block Layer

Transform Coefficients - TCOEFF - variable length
End of Block - EOB - 2 bits

Recommendation H.261 also includes information about error correction, video coding delay, video data buffering, etc, not presented here but that justify its lecture.

4.6 Results and comments

To have a clear idea of the global H.261 performance, often used as reference quality, some results are presented in the following for videotelephone (VT) sequences - *CIF at 10 Hz* - and TV-like sequences - *CIF at 25 Hz*. The sequences are the ones internationally used for performance comparisons, namely: *C-"Claire" (photo 1), M-"Miss America", T-"Trevor", VTPH-"C+M+T", D-"Diva", FG-"Flower Garden", P-"Popple", R-"Renata", TT-"Table Tennis" (photo 2) and TV-like-"D+FG+P+R+TT"*. Sequences are coded with suitable bit rates with and without motion compensation, e.g. 64 and 128 kbit/s for the VT sequences and 1024 and 2048 kbit/s for the TV-like ones.

photo 1 Typical videotelephone frame - "Claire"

The statistics shown do not consider the first frame since it requires a particular processing; this frame is usually coded with a fixed QS, starting the second frame coding with half full buffer as advised by the Reference Model. Moreover all the MC macroblocks are filtered which means that all the 'MC not filtered' macroblock classes are not used. The following abbreviations are used in the tables:

- **Rate** - *Bit rate in kbit/s*
- ***RVar F*** - *Bit Rate Variance (frame level) in (bit/s)^2*
- ***Bits Y/L, U, V %*** - *Percentage of Luminance or Chrominance data considering all the bits but the header bits*

- ***MVD %*** *- Percentage of motion vectors data considering all the bits but the header bits*
- ***EOB %*** *- Percentage of EOB and CBP data considering all the bits but the header bits*
- ***ATT %*** *- Percentage of MTYPE, MBA and MQUANT considering all the bits but the header bits*
- ***PMBR F*** *- Peak to Mean Bit rate Ratio (frame level) considering all the bits*
- ***Class X MB's*** *- Average number of class X MB's per frame*
- ***XSNR - 1/2*** *- Average X Signal to Noise Ratio - frame level (dB) - using/non using motion compensation*
- ***Min/Max LSNR - 1/2 F/G*** *- Minimum/maximum Luminance Signal to Noise Ratio using/non using motion compensation - frame (F) or GOB (G) level*
- ***QS 1/2*** *- Average Quantization Step - frame level - using/non using motion compensation*
- ***LVar F/G*** *- LSNR Variance - frame or GOB level*

photo 2 Typical TV-like frame - "Table Tennis"

The Signal to Noise Ratio is always computed as

$$SNR = 20 \log (255/rmse)$$

where

$$rmse = SQR\left[\sum_{i=1}^{n1}\sum_{j=1}^{n2}\frac{\left[f(i,j)-f'(i,j)\right]^2}{n1\ n2}\right]$$

f(i,j) is the original sample value
f'(i,j) is the coded sample value
n1, n2 are the matrix dimensions

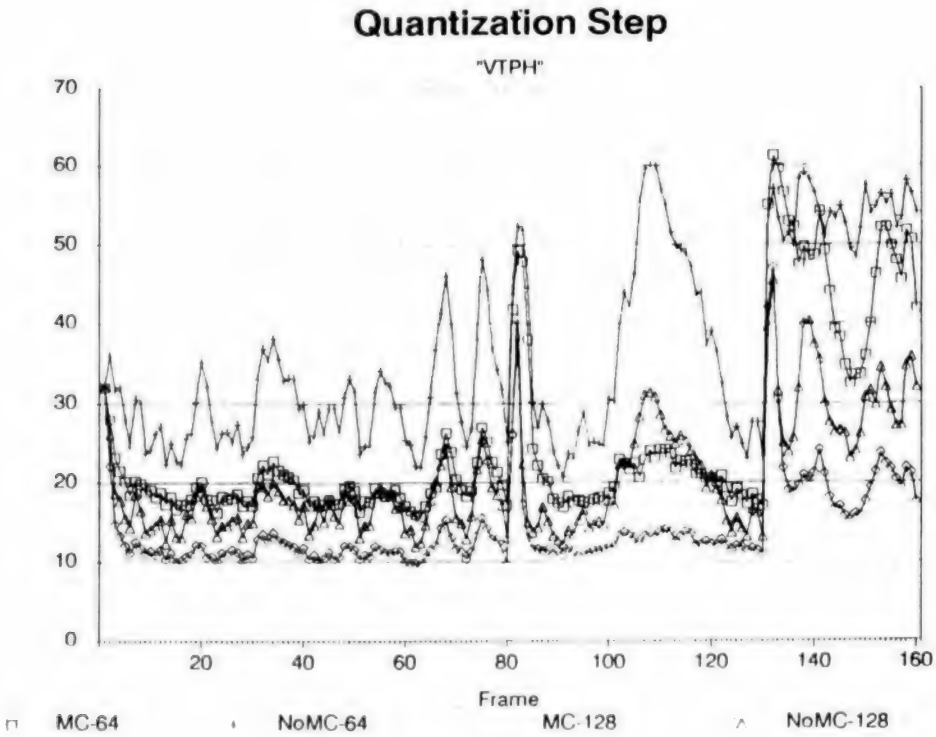


figure 14 Quantization step

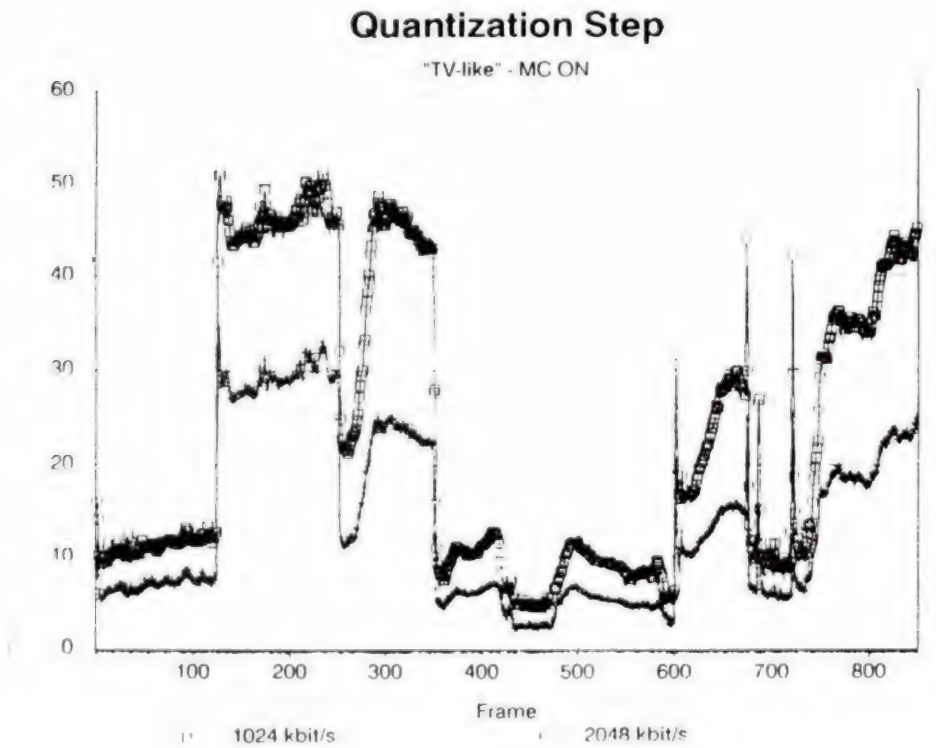


figure 15 Quantization step

Table 1
Some H.261 quality statistics - VT sequences

Sequence	C	C	M	M	T	T	VTPH	VTPH
Rate	64	128	64	128	64	128	64	128
LSNR 1	38.0	40.2	37.8	39.2	33.2	35.7	36.4	38.9
LSNR 2	35.8	38.4	35.6	37.8	28.8	34.2	34.0	37.3
USNR 1	39.0	40.5	37.7	38.8	39.0	40.5	37.9	39.8
USNR 2	37.6	39.4	36.6	37.8	38.4	39.8	36.9	38.8
VSNR 1	42.6	43.8	38.3	40.1	40.9	42.4	39.9	42.1
VSNR 2	41.2	42.8	36.1	38.5	40.2	41.7	38.6	41.0
Min 1 LSNR F	36.4	37.6	36.8	37.6	31.2	34.4	16.8	26.1
Min 2 LSNR F	33.8	36.7	31.1	36.1	23.0	33.1	17.9	28.2
Max 1 LSNR F	39.0	40.9	38.5	39.7	34.1	36.5	39.0	40.9
Min 1 LSNR G	34.4	35.9	34.3	36.1	24.8	31.1	12.7	16.3
Min 2 LSNR G	31.3	33.9	24.4	33.5	15.4	28.9	12.7	19.0
Max 1 LSNR G	45.2	46.6	42.9	43.6	36.6	40.1	45.2	46.6
QS 1	19.4	11.4	20.2	12.4	40.4	19.5	25.4	14.0
QS 2	29.8	16.8	35.3	19.3	52.2	29.8	36.6	20.4
LVar F	0.2	0.3	0.2	0.1	0.4	0.2	12.8	4.2
LVar G	8.7	6.8	4.2	3.7	4.8	4.0	15.9	8.9

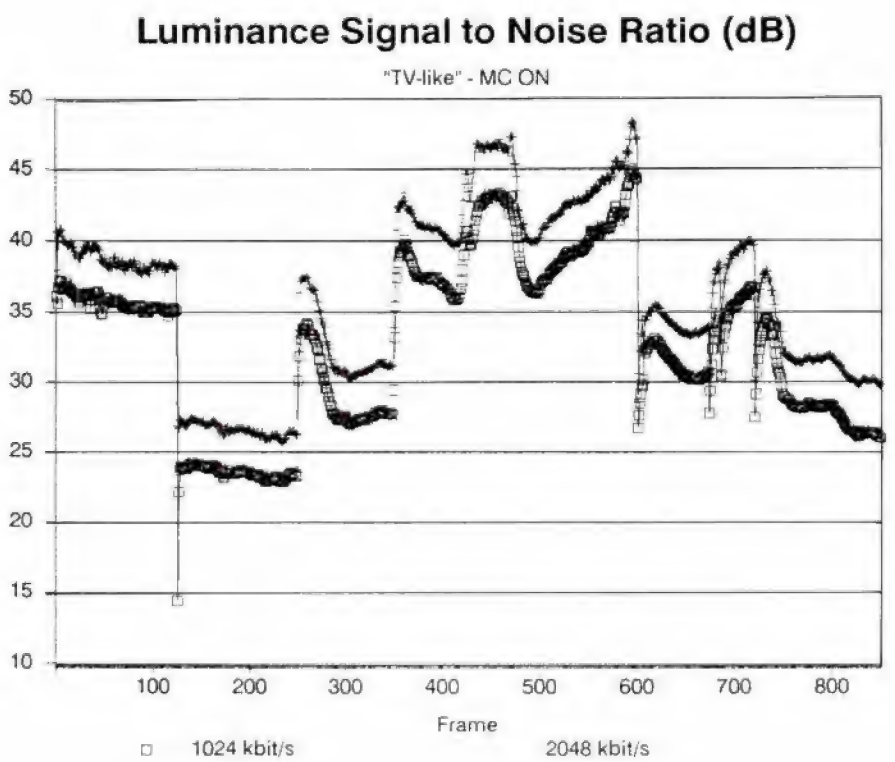


figure 13 Luminance Signal to Noise Ratio

The results obtained, which do not amount to a detailed H.261 performance analysis, allow to conclude:

- *The subjective and objective qualities vary along the sequences depending on the image activity and on the available data resources. The negative impact of this compromise is higher for TV-like sequences where the activity variations are higher (figures 12 and 13). Image activity is related to image variability both in space and time.*

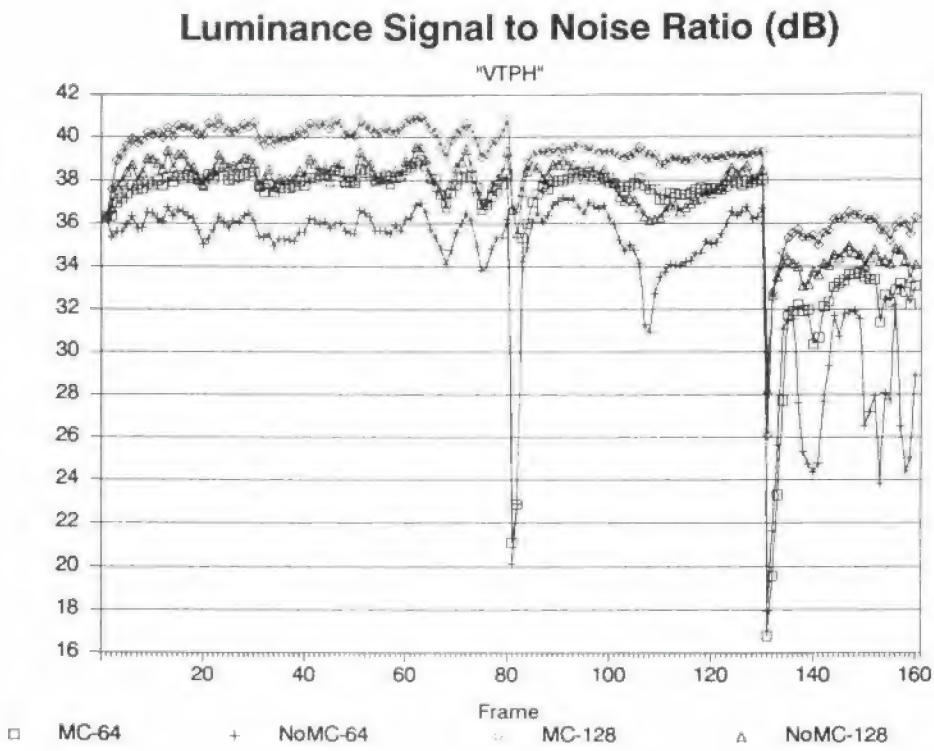


figure 12 Luminance Signal to Noise Ratio

- *The selected Quantization Step Coding Control proved to be efficient, reaching stationary equilibrium quickly (figures 14 and 15).*

- *The peak to Mean Bit rate Ratio (frame level) is very close to unity due to the tight coding control even if some variation may be absorbed by the coder buffer (PMBR<2), specially on scene cuts (figures 16 and 17).*

- *The subjective and objective qualities are critical for scene cuts and some particularly active pictures. For scene cuts and low bit rates, the buffer full block effects due to the limited resources are evident.*

- *The motion compensation performance improvement is important and reaches 2-4 dB on the global SNR averages; this improvement decreases with the bit rate (tables 1 and 2). The implemented motion compensation method is a block matching technique which may clearly detect a movement or only minimise the prediction error in the absence of a clear movement.*

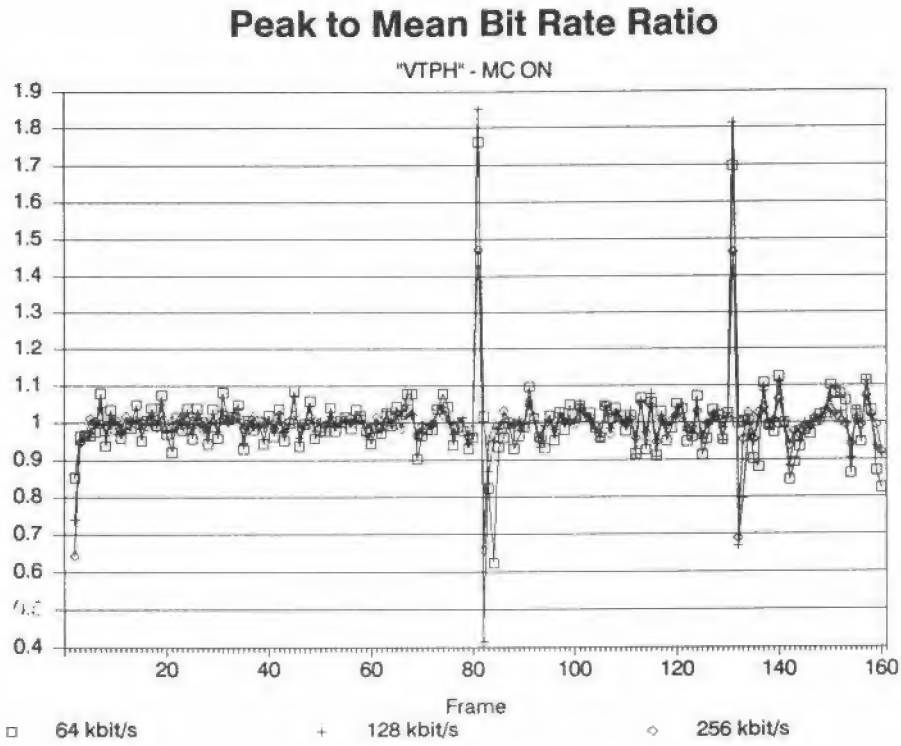


figure 16 Peak to Mean Bit Rate Ratio

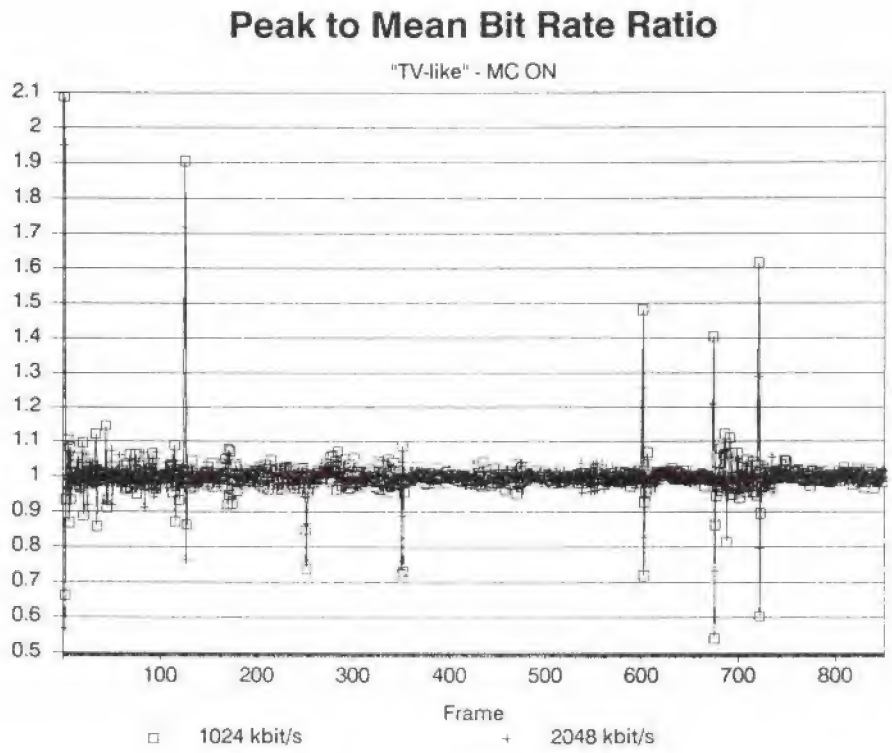


figure 17 Peak to Mean Bit Rate Ratio

Table 2
Some H.261 quality statistics - TV-like sequences

Sequences	D	D	FG	FG	P	P	R	R	TT	TT	TV	TV
Rate	1024	2048	1024	2048	1024	2048	1024	2048	1024	2048	1024	2048
LSNR 1	35.8	38.8	23.6	26.6	29.2	32.5	39.7	43.2	30.6	33.9	32.8	36.1
LSNR 2	35.3	38.4	18.3	24.1	27.7	31.8	36.5	40.1	29.0	32.5	30.4	34.3
USNR 1	35.3	38.1	26.8	28.3	29.4	33.1	42.5	45.1	34.9	37.1	35.3	37.8
USNR 2	35.1	37.9	24.3	26.9	28.1	32.4	40.5	42.8	34.1	36.2	33.9	36.6
VSNR 1	36.1	38.7	29.8	30.8	29.6	33.2	42.0	44.8	35.0	37.3	35.8	38.3
VSNR 2	35.9	38.5	29.4	29.8	28.3	32.5	39.8	42.4	34.2	36.4	34.6	37.0
Min 1 LSNR F	34.8	37.7	22.9	25.7	27.0	30.3	35.9	38.3	25.9	29.8	14.5	25.7
Min 2 LSNR F	34.2	36.8	17.7	23.5	23.9	29.3	31.8	35.6	22.1	27.6	14.4	23.5
Max 1 LSNR F	37.3	40.9	26.2	28.5	34.4	37.7	45.0	48.5	37.0	40.1	45.1	48.4
Min 1 LSNR G	30.2	33.3	20.2	22.9	25.7	29.0	33.0	33.7	23.5	27.1	7.0	22.7
Min 2 LSNR G	30.1	33.1	12.6	21.0	18.4	27.9	30.4	32.5	15.8	25.4	6.9	21.0
Max 1 LSNR G	47.0	47.6	35.7	37.0	41.6	43.2	48.1	50.1	43.6	45.0	48.2	49.9
QS 1	11.4	7.0	46.1	29.1	39.2	20.4	8.6	4.9	26.0	14.4	23.4	13.4
QS 2	11.7	7.1	57.8	45.8	51.7	25.4	14.3	7.5	35.8	19.4	31.1	18.7
LVar F	0.3	0.5	0.2	0.2	6.1	6.0	6.1	5.8	9.5	8.8	36.3	37.1
LVar G	22.1	11.5	10.1	8.4	10.2	10.1	9.8	8.2	15.8	12.9	44.0	41.2

- *The SNR chrominance performance is better than the LSNR performance due to the different compactness-energy properties of the two kinds of signals (tables 1 and 2). The chrominance, with a large number of very small coefficients, is less sensitive to the variation of the quantization step.*
- *The bit percentages and macroblock categories distributions depend on the video sequence activity characteristics and on the used data resources (tables 3 and 4).*

Table 3
Some H.261 bitstream statistics - VT sequences

Sequences	C	C	M	M	T	T	VTPH	VTPH
Rate	64	128	64	128	64	128	64	128
RVar F	8.4×10^6	2.4×10^7	8.5×10^6	3.8×10^7	3.6×10^7	5.1×10^7	4.3×10^7	2.2×10^8
Bits Y %	60.3	68.2	37.6	46.2	41.6	66.0	49.0	60.6
Bits U %	2.9	5.3	5.2	10.7	1.7	1.6	3.5	6.3
Bits V %	1.2	2.2	6.6	6.8	1.7	1.4	3.4	3.7
MVD %	6.7	2.8	12.5	5.6	22.4	9.1	11.8	4.9
EOB %	18.7	15.7	24.8	23.5	16.9	14.8	20.0	17.9
ATT %	10.2	5.8	13.2	7.2	15.6	7.1	12.3	6.5
PMBR F	1.09	1.08	1.08	1.09	1.18	1.12	1.76	1.85
PMBR G	2.35	2.61	2.56	2.29	2.98	2.66	6.41	4.04
Fixed MB's	268.7	207.1	207.6	127.2	195.7	193.2	234.4	178.4
Coded MC MB's	57.4	59.0	87.2	104.7	88.8	126.5	69.8	84.4
NoCod MC MB's	10.3	2.5	31.2	7.0	88.3	25.0	34.6	8.6
Coded NoMC MB's	59.3	127.1	69.9	157.0	11.9	39.1	53.5	120.1
Intra MB's	0.3	0.3	0.1	0.04	11.4	12.2	3.7	4.4

- *The subjective quality is usually acceptable for videotelephone sequences coded at 64 kbit/s. For TV-like sequences it is more difficult to define a lower data rate bound due to the very different picture activities - the ISO/MPEG algorithm uses around 1.1 Mbit/s for the video information but with a larger data rate variation and more powerful coding tools.*

Table 4
Some H.261 bitstream statistics - TV-like sequences

Sequence	D	D	FG	FG	P	P	R	R	TT	TT	TV	TV
Rate	1024	2048	1024	2048	1024	2048	1024	2048	1024	2048	1024	2048
RVar F	2.7×10^9	9.5×10^9	9.6×10^8	1.3×10^9	1.6×10^9	6.1×10^9	6.6×10^8	3.8×10^9	5.0×10^9	6.5×10^9	3.6×10^9	7.2×10^9
Bits Y %	72.8	71.8	82.8	88.2	53.7	60.2	75.3	80.0	81.2	84.5	75.2	79.0
Bits U %	9.6	12.0	1.9	2.8	13.8	15.1	3.4	4.7	2.5	3.8	5.1	6.4
Bits V %	9.3	10.7	0.4	0.7	14.1	15.0	4.5	5.8	3.1	4.1	5.3	6.3
MVD %	0.1	0.05	3.4	1.7	4.8	2.2	3.0	1.4	3.7	1.6	3.1	1.4
EOB %	6.5	4.5	8.6	5.1	10.7	6.1	11.0	6.8	6.5	4.5	8.6	5.5
ATT %	1.6	0.9	2.9	1.5	2.9	1.4	2.7	1.4	3.0	1.5	2.7	1.4
PMBR F	1.15	1.08	1.17	1.06	1.08	1.05	1.05	1.05	1.63	1.28	1.90	1.72
PMBR G	4.49	3.04	2.36	2.25	2.16	2.08	3.03	3.82	5.29	3.70	5.21	3.76
Fixed MB's	199.5	147.1	62.5	55.8	89.3	79.8	30.9	9.6	65.8	42.6	77.3	53.7
Coded MC MB's	12.3	9.1	276.1	293.9	226.9	226.7	225.2	229.4	136.2	172.9	174.7	188.8
NoCod MC MB's	0.6	0.1	28.8	13.7	8.8	0.9	10.8	3.4	97.0	53.6	37.2	18.8
Coded NoMC MB's	183.6	239.8	22.9	24.6	18.9	34.9	125.8	150.4	80.4	107.1	92.7	119.0
Intra MB's	0.0	0.0	5.6	8.0	52.1	53.6	3.2	3.2	16.6	19.7	14.1	15.6

5 Conclusions

Image will bring, in the years to come, relevant changes in the way we communicate. These changes are towards a more complete use of the human cognitive abilities where vision plays a primordial role. Videotelephone and videoconference have been proven to be useful and promising services through a number of recent experiments. These services are the top of the iceberg of a large number of video services, ready to spread or still being prepared.

The CCITT H.261 video coding algorithm is nowadays a fundamental element of video communications technology. Compatibility with this standard is thus a likely requirement for future coding algorithms. This fact was evident in the recent ISO standardisation process of the coding algorithm for moving images recording on digital storage media with a bit rate up to around 1.5 Mbit/s.

The image communication 'global village' is coming. Are we prepared for it?

References

[1] Pereira, F. (1991). Variable bit rate video coding, Ph.D. Thesis, Instituto Superior Técnico, Lisbon-Portugal.

[2] Pereira, F., Masera, L. (1990). Broadband video communications: the constant quality coding question, International Conference on Integrated Broadband Services and Networks, London-UK.

[3] Pereira, F., Quaglia, M. (1990). Extension of the CCITT visual communication coding algorithm for operation in ATM networks, Image Communication, **1**, 2, 1, 13-27.

[4] Habibi, A. (1977). Survey of adaptative image coding techniques, IEEE Transactions on Communications, **25**, 11, 1275-1284.

[5] Netravali, A., Limb, J. (1980). Picture coding: a review, Proceedings of the IEEE, **68**, 3, 366-406.

[6] (1982). Recommendation CCIR 601, Encoding parameters of digital television for studios .

[7] Netravali, A., Haskell, B. (1988). Digital Pictures - representation and compression, Plenum Press, New York.

[8] Hsing, R., Tzou, K. (1984). Video compression techniques: a review, GLOBECOM' 84, 1521-1526.

[9] Limb, J., Rubinstein, C., Thompson, J. (1977). Digital coding of color video signals - a review, IEEE Transactions on Communications, **25**, 11, 1349-1385.

[10] Chiariglione, L. (1986). La codifica del segnale video, Elettronica e Telecomunicazioni, 2, 43-68.

[11] (August 1990). CCITT SG VIII, ISO/Joint Photographic Experts Group, JPEG Technical Specification.

[12] (December 1990). CCITT SG VIII, ISO/Moving Pictures Experts Group, Committee Draft Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s.

[13] Grallert, H., Tengler, W., (1984).Image coding by Differential Pulse Code Modulation and transform coding techniques, a comparative survey, Siemens Forsch, 3, 95-99.

[14] Hammer, B., Tengler, W. (1984). Transform coding of digitized image signals, Siemens Forsch, 3, 105-108.

[15] Nasrabadi, N., King, R. (1988). Image coding using vector quantization: a review, IEEE Transactions on Communications, **36**, 8, 957-969.

[16] Linde, Y., Buzo, A., Gray, R. (1980). An algorithm for vector quantizer design, IEEE Transactions on Communications, **28**, 1, 84-95.

[17] Woods, J. O'Neil, S. (1986). Subband coding of images, IEEE Transactions on Acoustics, Speech and Signal Processing, **34**, 5, 1278-1288.

[18] Tseng, H., Fischer, T. (1987). Transform and hybrid transform/PCM coding of images using pyramid vector quantization, IEEE Transactions on Communications, **35**, 1, 79-86.

[19] Forcheimer, R. (1990). Some research topics in image coding, Image Coding Group, Linköping University, Sweden.

[20] (1990). P.OEM Fractal Images, Barnsley Communications, Inc, USA.

[21] Chiariglione, L. (1989). Trends in video communications, CSELT Technical Reports, **XVII**, 5, 349-356.

[22] Le Pannerer, Y. (1989). Video coding standards: position paper, RACE 1018-HIVITS Document, 1-8.

[23] (1990). Telettra Review - Special Issue for High Definition TV, **45**, 1.

[24] Kenyon, N.D. (1984). Videoconferencing, British Telecom Technol. Journal, **2**, 2, 5-18.

[25] (1984). CCITT SG XV, Series H.100 CCITT Recommendations, Characteristics of visual telephone systems.

[26] Okubo, S. (1989). Video codec standardization in CCITT study group XV, Image Communication, **1**, 1, 45-54.

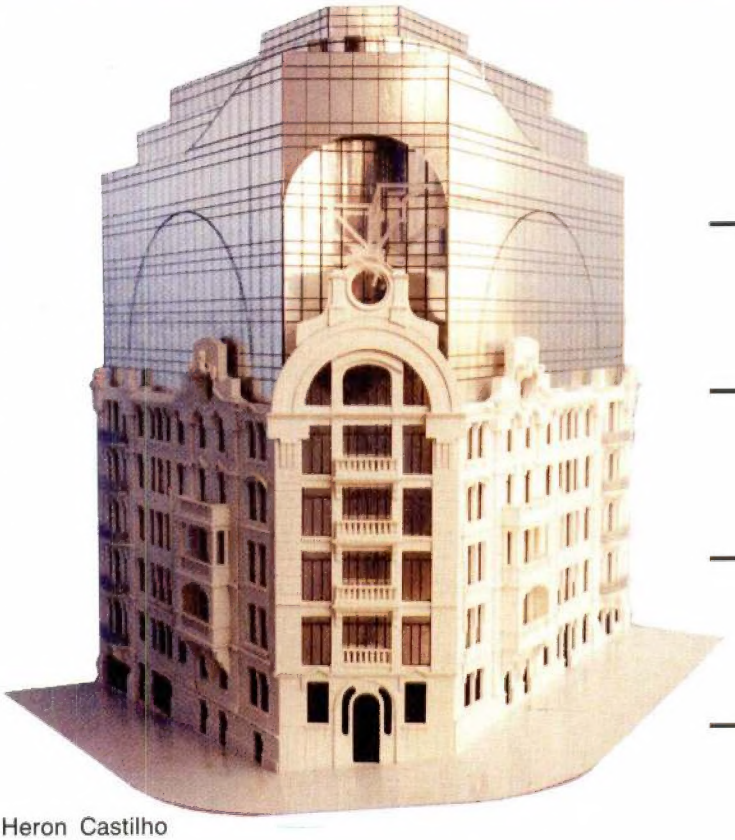
[27] (March 1990). CCITT SG XV, Draft Revision of Recommendation CCITT H.261, Video codec for audiovisual services at px64 kbit/s.

[28] Verbiest, W., Pinnoo, L., Voeten, B. (1988). The impact of the ATM concept on video coding, IEEE Journal on Selected Areas in Communications, **6**, 9, 1623-1632.

[29] (May 1989). CCITT SG XV - Specialists Group on Coding for Visual Telephony, Description of Reference Model 8 (RM8), COST 211 bis Simulation Group, Paris-France.

Sobre o Autor

Fernando M. Bernardo Pereira é Professor da Secção de Telecomunicações do Departamento de Engenharia Electrotécnica e de Computadores do Instituto Superior Técnico, Universidade Técnica de Lisboa.



Heron Castilho
- Lisboa

— Execução e Coordenação de Projectos Gerais

— Coordenação e Fiscalização de Obras

— Assessorias Tecno-Administrativas de Empreitadas

— Consultadorias Técnicas

ENPESIN Projectos e Gestão de Obras, Lda.

Av. Eng. Arantes e Oliveira, 5 - S/L-C 1900 Lisboa Telef.: 80 25 05 Fax: 80 27 03

Com o alto patrocínio do Sr. Eng. Luís Mira Amaral

PoSAT: Consórcio para o lançamento do 1.º satélite português

PoSAT:
Consórcio INETI, EFACEC, CPRM, ALCATEL, OGMA, UBI, IST, CEDINTEC

MARCONI Comunicações Globais

INETI

ALCATEL PORTUGAL

PEDIP

EFACEC

OGMA

eletrónica de potência, rectificadores, tokamak
power electronics, rectifiers, tokamak

João José Esteves Santana
Secção de Máquinas Eléctricas e Electrónica de Potência, Dep. Engª Electrotécnica e de Computadores, IST, UTL

A ELECTRÓNICA DE POTÊNCIA NA FÍSICA: O RECTIFICADOR DE ALIMENTAÇÃO DAS BOBINAS DO TOKAMAK DO CENTRO DE FUSÃO NUCLEAR DO IST

Resumo

Neste artigo é referida a importância da Electrónica de Potência na experimentação em Física. A Fusão Nuclear é um domínio onde sofisticados e potentes circuitos electrónicos de potência são utilizados.

O sistema de rectificação que alimenta as bobinas toroidais do TOKAMAK instalado no Centro de Fusão Nuclear do IST constitui o objectivo deste trabalho.

Justificada a escolha da associação em paralelo de rectificadores com retorno pelo neutro, é analisada detalhadamente a montagem em dupla estrela trifásica com primário em estrela. Os resultados são generalizados de modo a caracterizar montagem dodecafásica que constitui a alimentação do TOKAMAK.

Abstract

This article points out the major rule of Power Electronics in Physics experimentation. Nuclear Fusion is a domain where sophisticated and powerful power electronics are used.

The goal of this work is the rectifying system which supplies the TOKAMAK thoroidal inductances installed on the IST Nuclear Fusion Centre.

Justified the choice of parallel association of three-phase half wave rectifiers, one analyses in detail the three-phase double-star with star primary assembly. The results are generalised in order to characterise the dodeca-phase assembly which composes the TOKAMAK supplying system.

0 Introdução

A Electrónica de Potência é uma área do conhecimento que se caracteriza, actualmente, pela utilização de dispositivos semicondutores na conversão estática da energia eléctrica. Ela situa-se na encruzilhada dos principais domínios de aplicação técnica da electricidade que vão da electrotecnia ao desenvolvimento dos componentes e electrónica de comando, das máquinas eléctricas à utilização das modernas ferramentas de controlo, etc.

Os principais parâmetros que caracterizam uma fonte de energia eléctrica podem ser variados com circuitos electrónicos de potência: valor eficaz e valor médio da tensão ou da corrente, frequência, factor de potência, etc. Muitas vezes, estas operações destinam-se a assegurar o comando de grandezas não eléctricas, por exemplo o binário ou a velocidade dum motor ou a temperatura dum forno.

A Electrónica de Potência é necessária em todas as áreas da geração, transmissão, distribuição e aplicação da energia eléctrica. Praticamente, todos os bens de equipamento utilizam circuitos electrónicos de potência [1]: máquinas ferramentas (15%), locomotivas (15%), computadores (10% a 15%), entre parêntesis indica-se o custo relativo da parte da Electrónica de Potência.

A Física Experimental é, igualmente, um sector no qual o sucesso da sua actividade depende da facilidade de "manipulação" da energia eléctrica permitida pelos semicondutores de potência [2].

O desenvolvimento da Fusão Nuclear constitui um campo privilegiado de aplicação de sofisticados e potentes conversores electrónicos de potência. Neste trabalho, descreve-se o circuito de potência de uma fonte de corrente contínua que alimenta as bobinas que criam o campo magnético toroidal do TOKAMAK instalado no Centro de Fusão Nuclear do IST.

O circuito constituído pelas bobinas anteriormente referidas, é caracterizado por uma resistência de 10 mΩ e uma indutância de 1.88 mH. Este circuito deve ser alimentado por uma corrente contínua, cujo valor pode ser regulado entre 4000 A e 8000 A. Esta operação, que consiste na circulação da corrente contínua com um valor pré-determinado nas bobinas do TOKAMAK, deve durar tipicamente alguns segundos, após o que se segue um período "longo" de repouso. Deste modo, a fonte regulada de corrente contínua tem um funcionamento intermitente que no regime de pleno uso apresentará, por imposição construtiva devida à dissipação térmica, o seguinte ciclo: 3 s On / 3 min OFF.

Original submetido para publicação em 20 de Março de 1992

A Electrónica de Potência na Física:
O Rectificador de Alimentação das Bobinas do Tokamak do Centro de Fusão Nuclear do IST

1 Rectificador Controlado

A fonte que alimenta as bobinas do TOKAMAK, caracteriza-se por apresentar à saída uma tensão relativamente baixa (U<100V) e uma elevada corrente (4000A<I<8000A). Com estas especificações, a associação em paralelo de rectificadores com retorno pelo neutro é recomendável, pois apresenta, relativamente às montagens em ponte, as seguintes vantagens:

- a corrente que atravessa cada semicondutor é uma fracção da corrente total de saída, pois utilizam-se conversores em paralelo
- divide por dois as quedas de tensão nos semicondutores, pois utilizam-se rectificadores com retorno neutro.

Embora penalizando o dimensionamento do transformador, a associação em paralelo de rectificadores com retorno pelo neutro foi seleccionada pelas razões referidas. Assim, a figura 1 ilustra a montagem utilizada, ela consiste na associação em paralelo de duas associações em paralelo de rectificadores com retorno pelo neutro trifásicos, uma com o primário em estrela (Y) e a outra com o primário do transformador em triângulo (Δ). Este esquema apresenta as características de uma montagem dodecafásica, isto é, o índice de pulsação da tensão rectificada é p=12 e a corrente pedida à rede trifásica não apresenta, nomeadamente, as harmónicas de ordem 5 e 7.

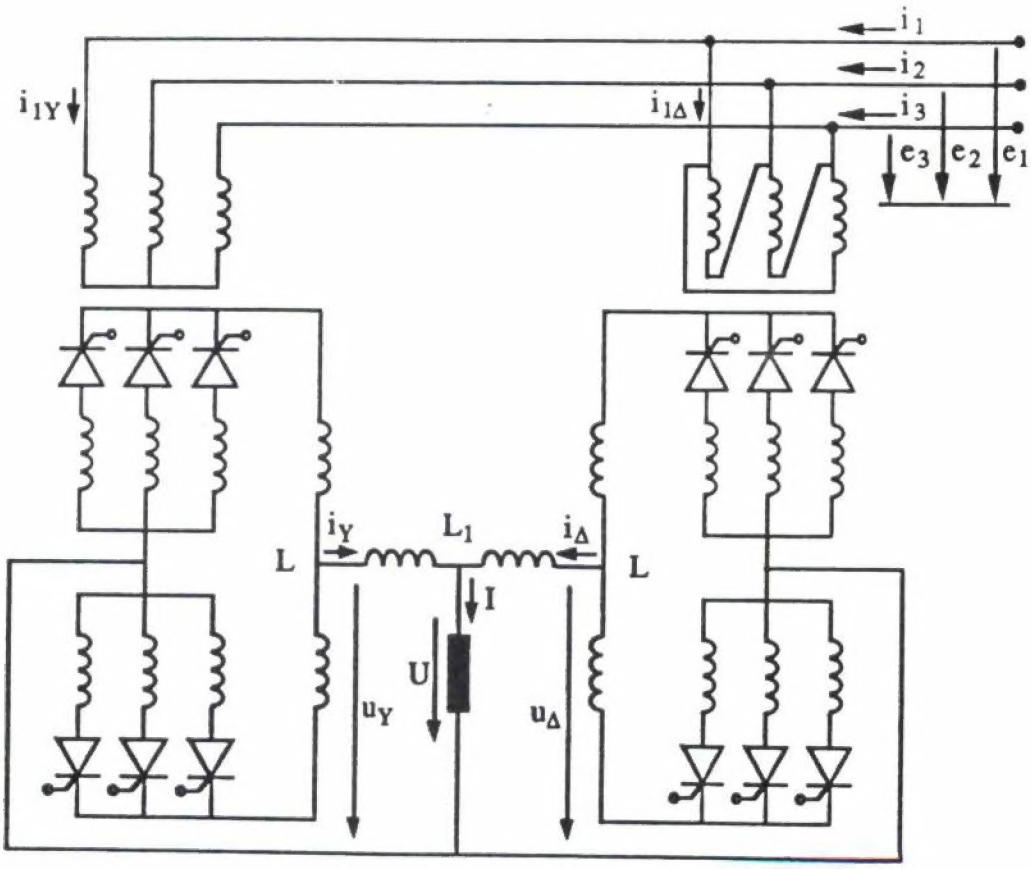


figura 1